A FINITE ELEMENT ANALYSIS OF THERMAL AND DEFORMATION PROCESSES IN METAL CUTTING

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

by
C MALLIKARJUNA SARMA

to the
DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
February, 1990

CERTIFICATE

This is to certify that the work entitled, "A Finite Element Analysis of Thermal and Deformation Processes in Metal Cutting" by C. Mallikarjuna Sarma has been carried out under my supervision and has not been submitted elsewhere for a degree.

T. Sundararajan 16.2.90

Dr. T. Sundararajan
Assistant Professor
Department of Mechanical Engineering
Indian Institute of Technology
Kanpur

IITK,
16th Feb. 1990

ACKNOWLEDGEMENTS

I must say I am extremely lucky to have the wonderful association with **Dr. T. Sundararajan** as my thesis supervisor. He has given a freehand in whatever I did but simultaneously was critical whenever I went astray. His inspiring guidance and constant encouragement throughout coupled with the invaluable suggestions helped me in gaining expertise in all aspects of Theoretical Modelling. I am indebted to him.

I am thankful to **Dr. G.K. Lal** for the useful discussions we had frequently and for his timely suggestions.

I am also thankful to my friends **Balaji** and **Lakshminarayana** for extending a helping hand at crucial times, and especially Mr. **J.P. Gupta** for doing the typing work.

Finally I thank all others who helped me in the project directly or indirectly.

IITK, **C. MALLIKARJUNA SARMA**
16 Feb. 1990

CONTENTS

ABSTRACT

A procedure for the application of the Finite Element technique to analyze the coupled stress, mass and heat balance euations in a two-dimensional Orthogonal metal cutting situation is presented. The material undergoing deformation is treated as a strain-rate sensitive, non-strain hardening visco-plastic solid. The visco-plastic behaviour is utilized to describe the plastic flow as equivalent to that of an incompressible, non-newtonian fluid. For the sake of simplicity, thermal softening is not taken into account. The resulting mass, momentum and heat transfer equations are solved iteratively with successive under-relaxation, using the Finite Element Method. The temperature and strain-rate distributions thus obtained are used in determining the shear plane and the dimensions of primary and secondary deformation zones. A parametric study has also been conducted to examine the effects of process variables such as the cutting velocity, chip thickness ratio and depth of cut on the temperatures at the chip-tool interface.This is because the high temperature distribution occurring in this region due to frictional heat dissipation is the prime reason for tool wear.

NOMENCLATURE

$\underset{\approx}{d}$, σ'_{ij}	Deviatoric Stresses
$\underset{\approx}{\sigma}$	Stress Tensor
$\underset{\approx}{\dot{e}}$	Strain Rate Tensor (per sec)
$\dot{\bar{e}}$	Strain Rate Invariant (per sec)
G	Modulus of Rigidity (N/m^2)
$\underset{\sim}{f}$	Body Force Vector (N/m^2)
p	Mean Normal Stress (N/m^2)
g	Acceleration due to gravity (m/s^2)
t	Depth of cut (mm)
t_c	Chip thickness (mm)
$\underset{\sim}{V}$	Velocity Vector (m/s)
V_c	Chip Velocity (m/s)
V_T	Tangential Velocity at chip-tool Interface (m/s)
u	Velocity in x direction (m/s)
v	Velocity in y direction (m/s)
x	Coordinate in cutting direction

y	Coorodinate normal to the cutting direction
ξ	Local x-coordinate
η	Local y-coordinate
α	Rake Angle (degrees)
$\emptyset$	Shear Angle (degrees)
μ	Dynamic Viscosity ($N-s/m^2$)
ρ	Density (Kg/m^3)
ψ	Stream Function (m^2/s)
ω	Vorticity (s^{-1})
σ_y	Uniaxial Yield Stress (N/m^2)
σ_{ij}	Stress component
e_{ij}	Strain component
$\bar{\tau}$	Effective yield shear stress (N/m^2)
τ_f	Effective frictional shear stress (N/m^2)
m	Fri ction factor
k	Thermal conductivity of workpiece material (W/mK)
k_t	Thermal conductivity of tool material (W/mK)
C_p	Specific Heat of workpiece material (KJ/KgK)
α	Thermal diffusivity (m^2/sec)
h	Overall heat transfer coefficient (W/m^2K)
Pe	Peclet Number

T	Absolute Temperature (K)
T_f, T_{amb}	Ambient Temperature (K)
$\dot{Q}$	Volumetric Heat generation rate (W/m^3)
$\dot{Q}_f$	Frictional Heat generation rate (W/m^3)

OPERATORS

L	A differential operator
$\underset{\sim}{\nabla}$	Gradient operator
∇^2	Double differential operator
∂	Partial differential operator

SUPERSCRIPT

*	Indicates non-dimensionalization

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 GENERAL BACKGROUND

The importance of machining in the modern engineering industry need not be over-emphasised when it is stated that nearly half of the engineering products are produced through machining at one stage or the other. In this age of wide-spread automation, we are witnessing the trend that the demand for productivity and process optimization is increasing day by day. As the search for new materials and processes is continuing, effort is still underway to improve conventional processes by reducing the material wastage to a bare minimum, increasing tool life and improving the product finish. In view of the crucial role played by machining among all the material-processing operations, a scientific understanding of various metal-cutting processes is vital for modern engineering technology and practice.

A detailed analysis of the mechanics of metal-cutting which sheds light on the underlying plastic deformation processes and stresses is very helpful in the prediction of some of the input parameters such as the cutting forces. The evaluation of the cutting forces, in turn, enables one to determine the power input and to choose the tool material. Further, minimization of the power input for a certain material removal rate (MRR) often forms the objective of optimizing the metal-cutting operation. Also, an

insight into the effects of various process variables like the tool geometry, tool material properties, cutting velocity and tool-tip temperature is useful in determining the optimal cutting conditions, for lowering machining costs and increasing the productivity.

Over the past few decades, a large volume of experimental data has been collected on various aspects of metal-cutting, but a thorough understanding of this highly complex process is yet to emerge. Till date, a comprehensive theoretical treatment is also not available, though many an attempt has been made in the past in that direction. The present work, in its right earnest, is a small effort to fill some gaps existing in the theory of metal cutting. A coupled analysis of the deformation and thermal processes in the vicinity of the tool-tip has been attempted here, which is expected to provide an accurate estimate of the temperature field as well as the size of the plastic deformation zone.

1.2 REVIEW OF PREVIOUS WORK

Research in the area of metal cutting commenced right from the post-war period. Development both on the theoretical and experimental fronts took place gradually, with better theoretical models replacing their predecessors, as the computing and experimenting techniques advanced.

The first attempts towards developing an understanding of the Mechanics of Metal Cutting were performed by analysing the chip-formation process.As early as 1945, a pioneering work by Merchant[1]and Piispanen [2], was presented on the so-called

classical "single shear plane model". This model assumes that the chip is formed due to plastic deformation occuring along a single plane, known as the shear plane. The most advantageous aspect of this model is that it enables the determination of the average yield shear stress and the slip velocity along the shear plane, purely through simple geometric constructions. It also established beyond any doubt that metal cutting is basically a shear deformation process. The Merchant Piispanen model was subsequently improved by Palmer & Oxley [3] and Okushima & Hitomi [4]. These studies suggested that though the formation of the chip occurs due to the flow of the metal under shear, the deformation is not limited to a single plane; it occurs in a narrow region approximated by two parallel planes. This model later came to be known as the Thick Shear Zone Model. In view of the small thickness of the shear zone in comparison to its length, this model assumes that the state of stress within the deformation region is uniform simple shear.

Around the same period of the above-mentioned theoretical models efforts, were going on to validate these models with experimentation. Kececioglu [5] was the first person to observe the process through photo-micrographs in 1958. The plastic zone where the chip formation takes place, was photographed at short intervals using a quick-stop device. By noting the deformation of the grain boundaries, the size and the shape of the plastic deformation zone were estimated. This study established that the plastic region could be approximately represented as a thin parallel-sided zone. Later in 1969, a more effective way of using the photo-micrograph technique was proposed by Stevenson and Oxley

[6].Assuming a thin parallel-sided zone, they observed printed grids on the material, before and after the deformation. From the streamlines of metal flow, the strains and strain rates were calculated. The important process variable of flow stress (called by different names such as effective yield stress, dynamic shear stress (DSS) etc.) could also be evaluated, as a function of strain by this method, for a wide range of strain rates.Later investigators [7,8] proposed that the flow stress is a strong function of strain rate as well.

A totally different approach was proposed by J.T. Black [7], based on the theory of dislocation mechanics. It suggested, that the flow stress, is composed of two parts i.e. thermal and non-thermal. The observations of the study indicated that the plastic zone is divided into alternate shear and lamella bands. The shear bands (or shear fronts) give rise to the dominant non-thermal part while the lamella regions, lead to the thermal part of the flow stress. In this study, the flow stress was evaluated by relating the stress to the Stacking Fault Energy (SFE), which itself was obtained from the dislocation distribution in the metal. This approach, mostly used by metallurgists, has not progressed far and and it is difficult to apply this approach as long as the concepts of dislocation mechanics do not match exactly with those of continuum mechanics.

In all studies discussed above, the emphasis was only on the primary deformation zone occuring across the shear plane. It was Von Turkovich [8,9] who first observed the peculiar form of plastic flow, extremely localised in a small region in the chip,

ahead of the tool edge along the rake face. This was called the Secondary Shear Deformation Zone. The experimental studies conducted so far, have established that material behaves like a highly viscous fluid within the deformation zones and as a perfect solid (with infinite viscosity and zero strain rate) outside them. The viscosity, in such a fluid like statein fact, depends strongly on the material deformation rate.

With the advent of high-speed computers coupled with the increased awareness of the immense potential of numerical techniques like FDM and FEM considerable interest was devoted to the application of these procedures. The credit for the effective usage of the FEM technique to visco-plastic analysis goes to Zienkiewicz et al.[12,13].In 1973, these researchers applied FEM for the modelling of metal forming and extrusion problems. In metal cutting, the FEM technique was employed by Tay and Stevenson [14,15]for predicting the temperature field for orthogonal-cutting. They used hyperbolic streamlines in the primary deformation zone for calculating the velocity fields and strain-rates and derived semi-empirical relations for computing the heat generation. The only drawback of this study was that it did not provide a coupled analysis of the plastic deformation and thermal processes during metal-cutting.

In study similar to that of Tay & Stevenson[14,15], exhaustive experimentation backed by theoretical validation was conducted by Muraka & Hinduja[17]. These authors obtained the average chip-tool interfacial temperature for a wide range of cutting conditions. Balaji [18] applied the FEM technique to evaluate the temperature

field within the work-material and the tool, for coated carbide tools. He also measured the average tool-tip temperature by the tool-chip thermocouple method and found reasonable agreement with the FEM predictions.

The FEM solutions available at present for the metal-cutting situation provide only the temperature distribution in the chip and the tool for a given velocity field. In these analyses, the velocity fields themselves are obtained by empirical and geometrical considerations. The relations obtained from such considerations, however, suffer from gross simplifications which are necessitated by the complexities of the process as well as the domain. These simplifications, in turn, tend to alter the problem being solved considerably and also affect the accuracy of the solution.Also, the available temperature predictions through FEM have not been adequately compared with experimental results due to limitations in measuring the interfacial temperatures accurately.

1.3 OBJECTIVES & SCOPE OF PRESENT STUDY

In the present study, therefore, an attempt has been made to fill the gap in the analysis of metal-cutting by solving the coupled flow and energy equations. Stresses in the cutting zone have been modelled by considering visco-plastic material behaviour during deformation. Accordingly, the metal flow beyond the elastic limit is treated similar to that of a liquid of very high viscosity. The viscosity itself is described as a strong, non-linear function of the strain-rate and yield stress of the work-material. The material is considered to be purely strain-rate sensitive, with no effects of work-hardening or thermal softening.

A more general form of material deformation behaviour was not taken into account for the sake of simplicity.

It is well established from previous experiments that the deformation is restricted to a very small region around the tool tip. Therefore, a solution domain extending upto a few millimetres from the tool-tip into the workpiece and the chip has been considered. Beyond this domain, the material is assumed to be perfectly rigid. The tool is also taken to be perfectly rigid and sharp with no built-up edge. The tool edge is assumed to be orthogonal to the cutting direction. Further, for generating the domain and the finite element mesh easily, the geometry of the problem has been simplified in the following manner. The chip is considered to be bounded by plane parallel surfaces beyond the contact point. Adjacent to the shear plane, the free-surface of the chip is assumed to be part of a cylindrical surface with constant radius of curvature. Though the chip-curling process makes the real boundaries of the problem very complex, the simplifications invoked in the present study are appropriate, as the focus of this investigation is to perform coupled stress and heat transfer analysis for a given domain.

Another important assumption invoked in the present study is that all the power expended in plastic deformation is converted into heat. Though some residual stresses/strains are retained both in the workpiece and the chip, this assumption simplifies computation to a great extent without sacrificing the accuracy of predictions, when compared with the experimental results.

The governing equations of flow and energy resulting after the incorporation of the above-mentioned assumptions, have been solved using the Finite Element Technique. The velocity, temperature and strain-rate distributions have been obtained and utilised to identify the shapes of isotherms and the plastic zone boundaries.

CHAPTER 2

THEORETICAL FORMULATION

In many ways, metal cutting is unique among plastic deformation processes. The geometry of the deforming material is unbounded in metal cutting, which is not the case for most metal forming operations. Also extremely localised asymmetric deformation occurs at exceedingly high strains and strain rates. The metal cutting action itself is a consequence of high compression followed by shearing of the metal. Not to mention the least, the fact that all the deformation phenomena occur in a minute volume of a few millimetres size, adds the new dimension of phenomenally high temperatures and temperature gradients in the neighbourhood of the tool tip. Consequently, any theoretical treatment of the metal cutting process is bound to possess certain relaxations and simplifications.

In the existing studies, simplifying assumptions are made with regard to the geometry of the deformation zone and the nature of stresses within it. These, along with the empirical determination of some of the process variables, have led to detailed calculations of the temperature field around the tool-tip. In the present study, the deformation mechanics has been treated in a fairly general manner and the temperature field predictions have been coupled with the deformation phenomena. Some of the involved features of metal-cutting, however, are not taken into account in the present study for the sake of simplicity.

2.1 PLASTIC DEFORMATION AND CHIP FORMATION IN ORTHOGONAL MACHINING

In metal-cutting, the fact that the width of chip is large compared to its thickness, renders the problem two-dimensional. This is especially so, when the tool edge is orthogonal to the cutting direction. In the present work, a two dimensional orthogonal machining process under dry cutting conditions is considered as shown in Fig. 2.1. The tool is assumed to be sharp, with no built-up edge. The workpiece is taken to be ductile so that it produces a continuous chip. The tool and the portion of the workpiece material outside the domain shown in Fig.2.1 are treated as perfectly rigid. Although the deformation of material in metal-cutting is similar to those of metal forming processes, a distinguishing feature is that the deformed material is separated from the parent workpiece in the form of a chip. Hence, the analysis of chip formation is tantamount to the study of plastic deformation phenomena in metal-cutting.

The chip-formation can be described in terms of the following sequence of events. The uncut material in motion is first interrupted by the cutting tool. The tool is shaped in a suitable way to cause a large compressive stress on the material. After sufficient compression, when the resulting shear stresses in the material attain the yield value, plastic flow of material occurs, in the direction of the shear forces. Finally, the already deformed material particles on the rake face, are considerably slowed down due to the frictional stress prevailing between the

chip and the tool and also the work-hardening of the deformed particles. Further more, since extremely high-temperature conditions are produced near the tool-tip, the deformed material particles tend to stick partially to the rake-face. However, these particles also slip past the rake-face, due to the push generated by the subsequent elements of deformed particles. In a continuous operation, these sequence of events occur progressively to each successive element of uncut material. These events result in the continuous formation of the chip which is eventually separated from the workpiece. The plastic deformation and the frictional rubbing of the chip on the tool cause enormous amount of heat generation. The high temperatures resulting from such heat production, considerably influenced the deformation process and also affect the tool wear. It is therefore important to study both the mechanics and thermal phenomena during the chip formation process.

Deformation mainly takes place in two regions, which are termed as primary and secondary zones. The region in which deformation occurs first is termed the Primary Shear Deformation Zone (PSDZ), wherein material is essentially sheared along a plane. The PSDZ is denoted by the region ABCD in Fig. 2.2. The mean width of PSDZ is an important criterion which gives an idea of the rate of deformation and the amount of heat generated. The narrower the region, the greater the rate of deformation. Further, this is believed to be dependent upon the cutting conditions. For instance, the mean width of PSDZ is generally taken to be inversely proportional to the cutting velocity.

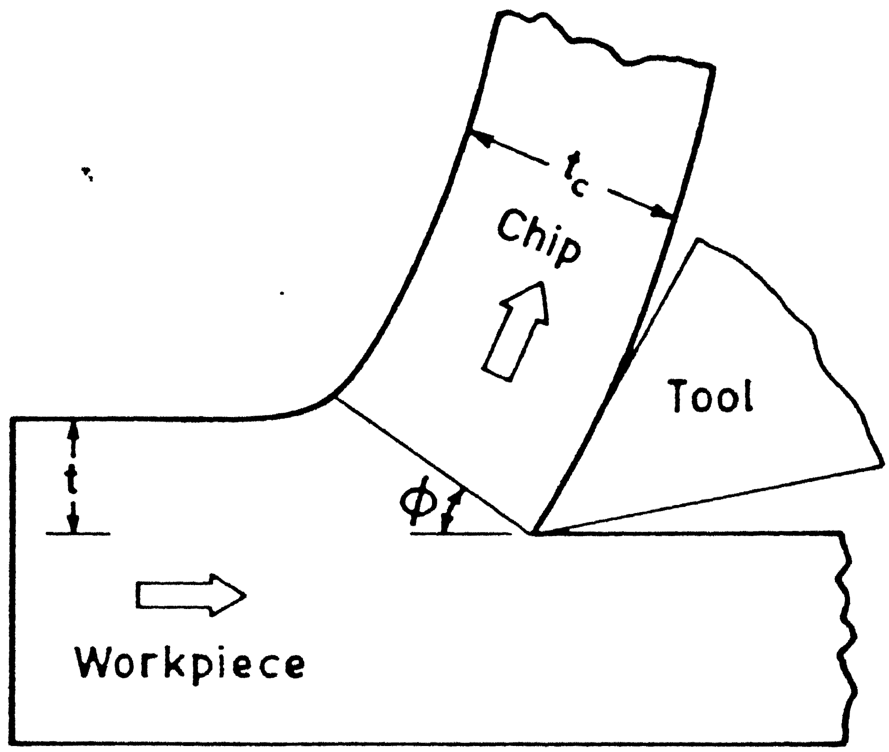


Fig. 2.1. Geometry of Chip Formation in Orthogonal Cutting.

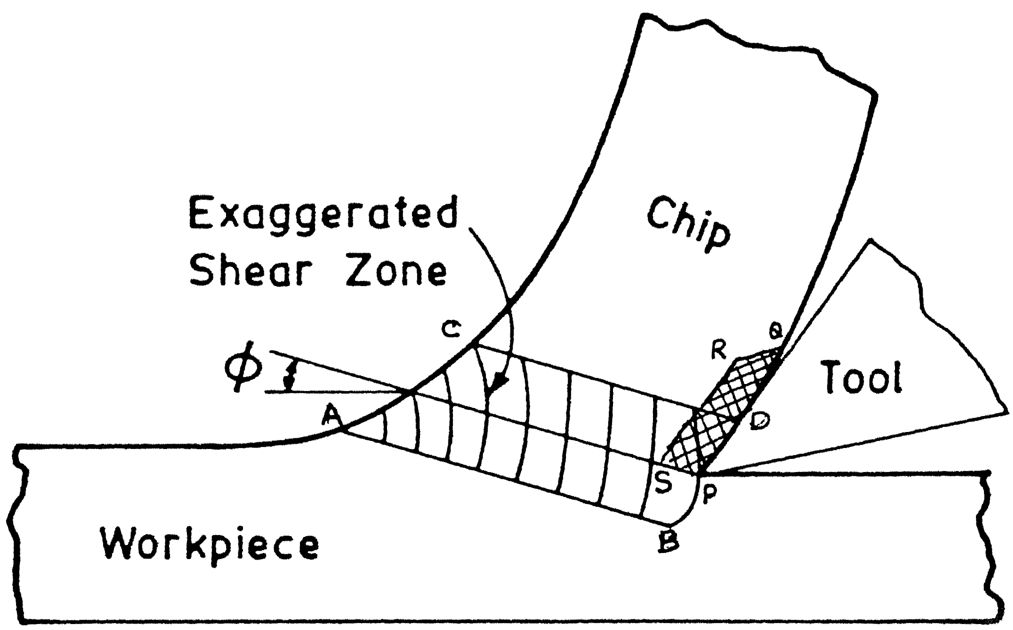


Fig. 2.2. Shear Zones During Metal Cutting.

A portion of the chip which lies adjacent to the contact patch between the chip and the tool undergoes additional deformation because of friction at the rake-face (area PQRS as shown in Fig. 2.2). The name Secondary Shear Deformation Zone (SSDZ) is given to this region. Each layer in SSDZ moves at a different speed because of shearing in a direction parallel to rake-face. Although this region is small as compared to PSDZ, the amount of heat produced is believed to be of a similar magnitude. An interesting point to note is that the highest temperature occurs on the interface between SSDZ and the rake-face. The temperatures in SSDZ are higher than those occuring in PSDZ mainly because the metal flow velocity is very low in the SSDZ and also the material which enters SSDZ has already been heated through PSDZ. At present, the thicknesses of SSDZ and PSDZ are estimated through empirical relations, which themselves are not on firm footing yet. It is definitely of interest to predict the extent of these zones, through a detailed analysis of the deformation phenomena.

2.2 VISCOPLASTICITY CONCEPTS

It is well known that under tensile load, a ductile material deforms elastically upto a certain strain, beyond which it starts yielding. This limiting strain is known as the Elastic limit. Upto the elastic limit, the stress and strain in the material are related by the Hooke's law. Beyond the elastic limit, when the material yields plastically, the stress (sometimes called as flow stress) becomes a function of the strain rate. Some of the properties of the solid undergoing plastic deformation are akin

to that of a Non-Newtonian fluid, in the sense that the relationship between the shear stress and the shear strain rate is non-linear. In fact, it is possible to define a pseudo-viscosity for a plastically yielding material which is the ratio between the shear stress and the shear strain rate. In the limit of a perfectly elastic or rigid solid, the viscosity assumes an infinitely large value.

It is often assumed in the analysis of plastic deformation processes that the density of the material is unaltered during the process. The material incompressibility which is implied by such an assumption, is reasonable since yield occurs primarily due to shear. Under such conditions, material volume hardly undergoes any change. Therefore, the sum of all the diagonal components of strain tensor (e_{ii}), which represents the volumetric strain, is zero. That is,

$$e_{ii} = e_{11} + e_{22} + e_{33} = 0 \qquad (2.1)$$

The strain tensor at a point itself can be expressed as

$$e_{ij} = \frac{1}{3} e_{ii} \delta_{ij} + e'_{ij} \qquad (2.2)$$

where

e_{ij} - strain tensor

e'_{ij} - deviatoric strain tensor

and $\delta_{ij} = 1$ if $i = j$

$= 0$ if $i \neq j$

Assuming material incompressibility to prevail, the various theories of plasticity developed so far attempt to relate the deviatoric part of strain (or its rate of change with respect to time) with the deviatoric stress. The deviatoric stress itself is defined in a manner similar to the deviatoric strain in eq. (2.2). Thus,

$$\sigma'_{ij} = \sigma_{ij} - \frac{1}{3}\sigma_{ii}\,\delta_{ij} \qquad (2.3)$$

where

σ_{ij} - stress tensor

$\frac{1}{3}\sigma_{ii}$ - isotropic stress

σ'_{ii} - deviatoric stress.

In addition to finding the relationship between the deviatoric stress and deviatoric strain, it is important todefine the criterion for material yield itself. An extensively used yield criterion in the analysis of plastic deformation processes is due to Von Mises. According to this criterion the second invariant of deviatoric stress tensor (represented by J_2) determines the condition of yielding. Mathematically, yield occurs when

$$J_2 = \frac{1}{2}\sigma'_{ij}\,\sigma'_{ij} \geq K^2 \qquad (2.4)$$

where K is a material yield constant.

Since material is said to be plastically deformed when yielding occurs, the total strain deviation is composed of two

parts i.e. elastic and plastic. Thus, one can write,

$$e'_{ij} = e'^{(e)}_{ij} + e'^{(p)}_{ij} \qquad (2.5)$$

However, since the problems of metal forming and metal cutting involve huge plastic deformations, the elastic part of strain deviation is often neglected. This omission of elastic strain could introduce a maximum error of 2 to 3 percent which is permissible.

Hence,
$$e'_{ij} \approx e'^{(p)}_{ij} \qquad (2.6)$$

The total plastic strain $e^{(p)}_{ij}$ occuring for a loading period of time t is given by the area under the curve of $\dot{e}_{ij}$ versus time t. Thus,

$$e^{(p)}_{ij} = e^{(p)}_{ij}(o) + \int_o^t \dot{e}_{ij}\, dt \qquad (2.7)$$

where $e^{(p)}_{ij}(o)$ is the plastic strain at the initial time t=o.

As a consequence of the principle of material incompressibility (from eqns. 2.1 and 2.6), as

$$e^{(p)}_{ii} = o \qquad (2.8)$$

the strain rate tensor $e^{(p)}_{ij}$ in equation 2.7 is therefore, same as that of the pl astic deviation strain tensor $e'^{(p)}_{ij}$.

For varying loading sequence, equation (2.8) for the total plastic strain $e_{ij}^{(p)}$ can often be replaced by an algebraic sum of the infinitesmal strain increments. The addition of all of the minute strain increments to obtain the total strain forms the essence of the Incremental theory of plasticity.

2.3 MATHEMATICAL FORMULATION

A. CONSTITUTIVE RELATION FOR VISCO-PLASTIC FLOW

In a metal cutting problem, it is conceivable to describe the metal flow, as that of a Von-Mises type of strain rate-sensitive, non-strain hardening,visco-plastic material, this, in turn, is akin to an incompressible (constant in density), non-newtonian fluid, as discussed earlier. An important aspect to be noted here is that the viscosity of such a "fluid" is dependent on the current local strain rates, total accumulated plastic strain undergone by the material and the local temperature. In the present analysis, however, work-hardening and thermal softening characteristics are assumed to be absent. Although the assumption has been invoked here for the sake of simplicity, a weak justification for the same can be offered based on the fact that work-hardening and thermal softening tend to cancel each other. Therefore, the material has been assumed to be purely rate-sensitive.For such a situation, the constitutive relation is expressed in the form :

$$\sigma'_{ij} = 2 \mu \dot{e}'_{ij} \tag{2.9}$$

where $\dot{e}'_{ij}$ is the derivatoric part of strain-rate tensor. The strain-rate tensor may itself be expressed in terms of the velocity gradients in the material. Its components in two dimensions (for orthogonal cutting) are given by

$$\dot{e}_{11} = \frac{\partial u}{\partial x} \tag{2.10a}$$

$$\dot{e}_{22} = \frac{\partial v}{\partial y} \tag{2.10b}$$

$$\dot{e}_{12} = \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) = \dot{e}_{21} \tag{2.10c}$$

where u and v are the velocity components in x and y directions.

The viscosity μ for a strain-rate sensitive material is taken to be a function of uniaxial yield stress of of the material (σ_y) and the strain-rate invariant ($\bar{\dot{e}}$). Mathematically,

$$\mu = f(\sigma_y, \bar{\dot{e}}) \tag{2.10}$$

In the present study, the constitutive relations of the form

$$\mu = \frac{\sigma_y + \left(\frac{\bar{\dot{e}}}{\sqrt{3}\,\gamma}\right)^{1/n}}{\sqrt{3}\,\bar{\dot{e}}} \tag{2.11a}$$

has been considered where γ and n are the physical constants which define the visco-plastic characteristics of the material. They usually take values between 1 and 2[See Appendix B].

Equation 2.11a can be looked at from another angle. As stated in Section 2.2, for a plastically deforming material, yield is a function of strain-rate ($\dot{\bar{e}}$) as well as the uniaxial yield stress (σ_y). This stress, often called as flow stress or the effective yield stress and denoted by $\bar{\sigma}$, is the term in the numerator of eq. 2.11a i.e.

$$\bar{\sigma} = \sigma_y + \left\{ \frac{\dot{\bar{e}}}{\sqrt{3}\,\gamma} \right\}^{1/n} \qquad (2.11b)$$

$$= f(\sigma_y \,,\; \dot{\bar{e}})$$

Rearranging terms in eq. 2.11a gives

$$\bar{\sigma} = \sqrt{3}\,\mu\,\dot{\bar{e}} \qquad (2.11c)$$

It may be noted that eq. 2.11c can also be derived from the Von-Mises criterion for yield and the stress-strain relationship of eq. 2.9. The strain-rate invariant $\dot{\bar{e}}$ can be expanded in terms of the strain-rate tensor as

$$\dot{\bar{e}} = \sqrt{2\,\dot{e}_{ij}\,\dot{e}_{ij}} \qquad (2.12a)$$

In a two-dimensional situation, the above equation reduces to

$$\dot{\bar{e}} = \sqrt{2(\dot{e}_{11}^{2}+\dot{e}_{22}^{2}+2\dot{e}_{12}\dot{e}_{21})} \qquad (2.12b)$$

In terms of velocity derivatives, $\dot{\bar{e}}$ can be expressed in a form

$$\dot{e} = \sqrt{2\left\{\left(\frac{\partial u}{\partial x}\right)^2 + \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2\right\}} \qquad (2.12c)$$

B. GOVERNING EQUATIONS

The necessity of considering the metal cutting problem in a different perspective from the earlier studies was highlighted in Chapter 1. Especially, the size and shape of the plastic zones and the temperature in the vicinity of the tool tip need to be predicted, without any adhoc assumptions or use of empirical data. Needless to say, this calls for a simultaneous application of mechanics of metal cutting and heat transfer theory. From the considerations of material balance, stress balance and heat balance, the governing equations in steady state for the velocity, pressure and temperature fields are given by :

$$\underset{\sim}{\nabla} \cdot \underset{\sim}{V} = 0 \qquad \text{(From Mass Balance)} \qquad (2.13)$$

$$\rho(\underset{\sim}{V} \cdot \underset{\sim}{\nabla} \underset{\sim}{V}) = \underset{\sim}{\nabla} \cdot \underset{\approx}{\sigma} + \underset{\sim}{f} \qquad \text{(From Stress Balance)} \qquad (2.14)$$

and

$$k \nabla^2 T + \dot{Q} = \rho C_p (\underset{\sim}{V} \cdot \underset{\sim}{\nabla} T) \qquad \text{(From Energy Balance)} \qquad (2.15)$$

where, ρ = Density of workpiece material

$\underset{\sim}{V}$ = The Velocity Vector

$\underset{\approx}{\sigma}$ = Stress tensor

$\underset{\sim}{f}$ = Body force vector

k = Thermal conductivity of work-material

T = Temperature

$\dot{Q}$ = Heat Generation Rate per unit volume

C_p = Specific heat of workpiece material.

FLOW EQUATIONS

From equation (2.4b), the stress tensor in the Momentum Equation can be expressed as

$$\underset{\approx}{\sigma} = -p \underset{\approx}{I} + \underset{\approx}{d} \qquad (2.16a)$$

where, p is the pressure and, $\underset{\approx}{d}$ is the deviatoric part of stress tensor.

Also the deviatoric stress tensor $\underset{\approx}{d}$ can be expressed in terms of the velocity gradients in the form :

$$\underset{\approx}{d} = \mu \; (\underset{\sim}{\nabla} \underset{\sim}{V} + \underset{\sim}{\nabla} \underset{\sim}{V}^T) - \frac{2}{3} \mu(\underset{\sim}{\nabla} . \underset{\sim}{V}) \underset{\approx}{I} \qquad (2.16b)$$

where $\underset{\sim}{\nabla} \underset{\sim}{V}^T$ represents the transpose of $\underset{\sim}{\nabla} \underset{\sim}{V}$ matrix. Simplifying eq.2.16b through the incorporation of material incompressibility condition and substituting for $\underset{\approx}{d}$ in eq. 2.16a, one obtains

$$\underset{\approx}{\sigma} = -p \underset{\approx}{I} + \mu \; (\underset{\sim}{\nabla} \underset{\sim}{V} + \underset{\sim}{\nabla} \underset{\sim}{V}^T) \qquad (2.16c)$$

Incorporating the above relation in the Momentum Equation (2.14), gives

$$\rho(\underset{\sim}{V} \cdot \underset{\sim}{\nabla} \underset{\sim}{V}) = \underset{\sim}{\nabla} \cdot \left\{ -p \underset{\approx}{I} + \mu(\underset{\sim}{\nabla} \underset{\sim}{V} + \underset{\sim}{\nabla} \underset{\sim}{V}^T) \right\} \tag{2.17}$$

where the body force vector $\underset{\sim}{f}$ (which usually consists of the weight of material per unit volume) has been absorbed into the pressure.

COMPONENT FORM OF FLOW EQUATIONS

As discussed earlier, the problem can be considered as a two-dimensional one, since the width of the chip is large in comparison to the thickness. The material balance equation for the two-dimensional situation can be represented as

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{2.18}$$

where u, v are the velocity components in x, y directions respectively (See Fig.2.3).

The stress tensor $\underset{\approx}{\sigma}$, in such a two-dimensional case, can be written in terms of its components as

$$\underset{\approx}{\sigma} = \sigma_{xx} \hat{i}\hat{i} + \sigma_{xy} \hat{i}\hat{j} + \sigma_{yx} \hat{j}\hat{i} + \sigma_{yy} \hat{j}\hat{j} \tag{2.19}$$

Substituting for $\underset{\approx}{\sigma}$ in the momentum equation (2.14), the x and y components of the momentum equations can be written in the form :

X-MOMENTUM

$$\rho(u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}) = \left\{\frac{\partial}{\partial x} (\sigma_{xx}) + \frac{\partial}{\partial y} (\sigma_{yx})\right\} \qquad (2.20a)$$

and

Y-MOMENTUM

$$\rho(u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y}) = \left\{\frac{\partial}{\partial x} (\sigma_{xy}) + \frac{\partial}{\partial y} (\sigma_{yy})\right\} \qquad (2.20b)$$

Again the components σ_{xx}, σ_{xy}, σ_{yx}, σ_{yy} of the stress tensor, can be obtained from eq. (2.16c) as :

$$\sigma_{xx} = -p + 2\mu \frac{\partial u}{\partial x} \qquad (2.21a)$$

$$\sigma_{xy} = \sigma_{yx} = (\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})\mu \qquad (2.21b)$$

and $$\sigma_{yy} = -p + 2\mu \frac{\partial v}{\partial y} \qquad (2.21c)$$

Substituting for these stress components in equations (2.20a) and (2.20b), the final forms of these momentum equations are given by

$$\rho(u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}) = \left\{\frac{\partial}{\partial x}(-p + 2\mu\frac{\partial u}{\partial x}) + \frac{\partial}{\partial y} \left[\mu(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})\right]\right\} \qquad (2.22a)$$

and

$$\rho(u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y}) = \left\{\frac{\partial}{\partial x}\ [\mu(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})] + \frac{\partial}{\partial y}\ (-p+2\mu\ \frac{\partial v}{\partial y})\right\} \quad (2.22b)$$

ENERGY EQUATION

In the energy equation, the important term to be carefully evaluated is the heat generation, since this can affect the temperature solution considerably. It is reasonable at this stage to postulate that the heat generated in metal cutting arises from two causes, namely : (1) plastic work in the PSDZ and (2) friction at the chip-tool interface. The above assumptions are reasonably valid, because they are the most plausible ways in which mechanical energy is converted into heat. More importantly, it is not improper to assume that all of the work done is converted into heat; the two important reasons for this are that the material density does not undergo a large change in value and also work hardening is not severe due to the high temperature conditions. Thus, the properties of the cut and the un-cut materials are not very different from each other. Keeping these factors in view, it appears reasonable to assume that all the mechanical power expended in machining is converted into heat.

Since the plastic power expended is given by the product of the flow stress and the strain rate, the volumetric heat generation rate can mathematically be expressed as

$$\dot{Q} = \bar{\tau}\ \dot{\bar{e}} \quad (2.23a)$$

where,

$\bar{\tau}$ = Effective yield shear stress or Flow stress
and, $\dot{\bar{e}}$ = Strain rate invariant.

The Von-Mises criterion allows the shear stress to be evaluated in terms of the effective uniaxial yield stress of the material as

$$\bar{\tau} = \frac{\bar{\sigma}}{\sqrt{3}} \tag{2.23b}$$

=> from eq. 2.11c $\quad \bar{\tau} = \mu \, \dot{\bar{e}}$ (2.23c)

Substituting the above expression for $\bar{\tau}$ in eq. 2.23a, the final expression for the rate of heat generation per unit volume is given by :

$$\dot{Q} = \frac{\bar{\sigma}}{\sqrt{3}} \, \dot{\bar{e}} \tag{2.23d}$$

The final dimensional form of energy equation can therefore be obtained from eq. 2.15 and 2.23d as

$$k \left[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right] + \frac{\bar{\sigma}}{\sqrt{3}} \, \dot{\bar{e}} = \rho C_p \left(u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} \right) \tag{2.24}$$

C. NON-DIMENSIONALIZATION OF FLOW EQUATIONS

As long as the number of parameters involved in a problem are few in number, it is realistic to stay with the dimensional form of variables, which enables ready comparison of predicted results with real values. However when the number of variables is large, analyzing the influence of each parameter on the results becomes an arduous task. The subject of metal cutting is known to involve parameters, which exhibit very complex inter-dependences among themselves. To circumvent such hardships and facilitate a detailed parameteric study, it is desirable to non-dimensionalize the variables of the problem with suitable scaling factors.

A real difficulty which often arises during non-dimensionalization, is the proper choice of a scaling factor for particular quantity. Though some general guidelines do exist, it can be stated that the primary guide in this process is intuition alone, since there is no unique way for non-dimensionalization. However, the best possible way is to give the physics of the problem a critical examination and incorporate as much of the physical characteristics of the problem as possible while selecting the scaling factors. For the present problem, to begin with, the velocity components u and v are non-dimensionalised with the cutting velocity (V); the distances with the depth of cut (t), and the pressure with the uniaxial yield stress (σ_y). The scaling factors for the remaining variables will be selected conveniently, during the course of non-dimensionalization itself. Thus, we define,

$$x^* = x/t;\ y^* = y/t;\ v^* = v/V;\ u^* = u/V \tag{2.25}$$

where the asterisk denotes a dimensionless quantity. Using the above definition in material balance equation,

$$\frac{\partial(u^* V)}{\partial(x^* t)} + \frac{\partial(v^* V)}{\partial(y^* t)} = 0 \tag{2.26}$$

one obtains

$$\frac{\partial u^*}{\partial x^*} + \frac{\partial v^*}{\partial y^*} = 0 \tag{2.27}$$

Similarly the x-momentum equation becomes

$$\frac{\rho V^2}{t}\left[(u^* \frac{\partial u^*}{\partial x^*} + v^* \frac{\partial u^*}{\partial y^*})\right] = \frac{V}{t^2}\left[\frac{\partial}{\partial x^*}(-\frac{t}{V}p + 2\mu\frac{\partial u^*}{\partial x^*}) + \frac{\partial}{\partial y^*}\{\mu(\frac{\partial u^*}{\partial y^*} + \frac{\partial v^*}{\partial x^*})\}\right] \tag{2.28a}$$

At this stage, the pressure and viscosity can be scaled as shown below :

$$\mu^* = \frac{\mu}{(\sigma_y t/V)}\ ;\quad p^* = p/\sigma_y \tag{2.28b}$$

Using the above expressions, the momentum equation can be simplified as

$$\frac{1}{\sigma^*}\left(u^* \frac{\partial u^*}{\partial x^*} + v^* \frac{\partial u^*}{\partial y^*}\right)$$

$$= \left[\frac{\partial}{\partial x^*}\left(-p^* + 2\mu^* \frac{\partial u^*}{\partial x^*}\right)\right] + \left[\frac{\partial}{\partial y^*}\left\{\mu^*\left(\frac{\partial u^*}{\partial y^*} + \frac{\partial v^*}{\partial x^*}\right)\right\}\right]$$

(2.29a)

where $\sigma^* = \dfrac{\sigma_y}{\rho V^2}$ (2.29b)

Similarly, the y-momentum equation can be reduced as

$$\frac{1}{\sigma^*}\left(u^* \frac{\partial v^*}{\partial x^*} + v^* \frac{\partial v^*}{\partial y^*}\right)$$

$$= \left[\frac{\partial}{\partial x^*}\left(\left\{\mu^*\left(\frac{\partial u^*}{\partial y^*} + \frac{\partial v^*}{\partial x^*}\right)\right\}\right)\right] + \left[\frac{\partial}{\partial y^*}\left(-p^* + 2\mu^* \frac{\partial v^*}{\partial y^*}\right)\right]$$

(2.29c)

It is important to note that the Viscosity equation is also to be normalized. Its non-dimensionalisation gives the following equation

$$\mu^* = \mu/(\sigma_y\, t/V) = \left[\frac{\sigma_y + \left(\frac{\dot{e}}{\sqrt{3}\,\gamma}\right)^{1/n}}{\sqrt{3}\ \dot{e}}\right] \times \frac{1}{(\sigma_y t/V)}$$

(2.30)

$$\mu^* = \frac{1+\frac{1}{\sigma_y}\left\{\frac{\dot{\bar{e}}^*}{\gamma\sqrt{3}}\left(\frac{V}{t}\right)\right\}^{1/n}}{\sqrt{3}\ \dot{\bar{e}}^*}$$

or,

$$\mu^* = \frac{1 + B^*\ (\dot{\bar{e}}^*)^{1/n}}{\sqrt{3}\ \dot{\bar{e}}^*} \tag{2.31a}$$

where

$$B^* = \frac{1}{\sigma_y}\left[\frac{V}{\sqrt{3}t\gamma}\right]^{1/n} \tag{2.31b}$$

NON-DIMENSIONALIZATION OF ENERGY EQUATION.

The energy equation in dimensional form as given by eq. 2.24 is

$$k\left[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right] + \frac{\bar{\sigma}}{\sqrt{3}}\ \dot{\bar{e}} = \rho C_p\ \left(u\ \frac{\partial T}{\partial x} + v\ \frac{\partial T}{\partial y}\right)$$

The normalization of temperature T in this equation can be done using some reference temperature T_{ref}, as

$$T^* = \frac{T}{T_{ref}} \tag{2.32a}$$

T_{ref} can be conveniently chosen as shown below :

Substituting for T and the other quantities in terms of their dimensionless counterparts, leads to

$$\frac{kT_{ref}}{t^2}\left[\frac{\partial^2 T^*}{\partial x^{*2}} + \frac{\partial^2 T^*}{\partial y^{*2}}\right] + \frac{V\sigma_y}{t}\,\frac{\sigma^*}{\sqrt{3}}\,\dot{\bar{e}}^*$$

$$- \rho C_p\,\frac{VT_{ref}}{t}\,(\,u^*\,\frac{\partial T^*}{\partial x^*} + v^*\,\frac{\partial T^*}{\partial y^*}\,) = 0 \qquad (2.32b)$$

Multiplying by $t^2/(\rho C_p)$, the above equation reduces to

$$\frac{k}{\rho C_p}\,T_{ref}\left[\frac{\partial^2 T^*}{\partial x^{*2}} + \frac{\partial^2 T^*}{\partial y^{*2}}\right] + Vt\,\frac{\sigma_y}{\rho C_p}\,\frac{\sigma^*}{\sqrt{3}}\,\dot{\bar{e}}^*$$

$$- Vt\,T_{ref}\,[\,u^*\,\frac{\partial T^*}{\partial x^*} + v^*\,\frac{\partial T^*}{\partial y^*}\,] = 0 \qquad (2.32c)$$

$$\left.\begin{array}{l}\text{Taking } \sigma_y/(\rho C_p) = T_{ref} \text{ and} \\ \\ \qquad k/(\rho C_p) = \alpha \text{ (thermal diffusivity)}\end{array}\right\} \qquad (2.32d)$$

the final form of dimensionless heat balance equation becomes

$$\left[\frac{\partial^2 T^*}{\partial x^{*2}} + \frac{\partial^2 T^*}{\partial y^{*2}} \right] + Pe \frac{\sigma^*}{\sqrt{3}} \bar{\dot{e}}^* - Pe \left[u^* \frac{\partial T^*}{\partial x^*} + v^* \frac{\partial T^*}{\partial y^*} \right] = 0$$

(2.33a)

where $Vt/\alpha = Pe$ (Peclet number) (2.33b)

2.4 BOUNDARY CONDITIONS

The modelling of the boundary conditions is equally important as the modelling of differential equation themselves, since different boundary conditions may yield drastically different results. In fact, experience reveals that the toughest part of the problem formulation is the prescription of proper boundary conditions. Hence this section denotes due attention to the formulation of each boundary condition.

To facilitate easy computation and to implement the boundary conditions in a systematic fashion, it is desirable to divide the boundary surface into individual segments, each with a different type of boundary condition. This division is essential particularly when the domain is complicated. For the present study, the total boundary has been divided into eight separate segments as shown in Fig. 2.3.

The first three surfaces I, II and III which form the boundary in the workpiece material are essentially quite far off

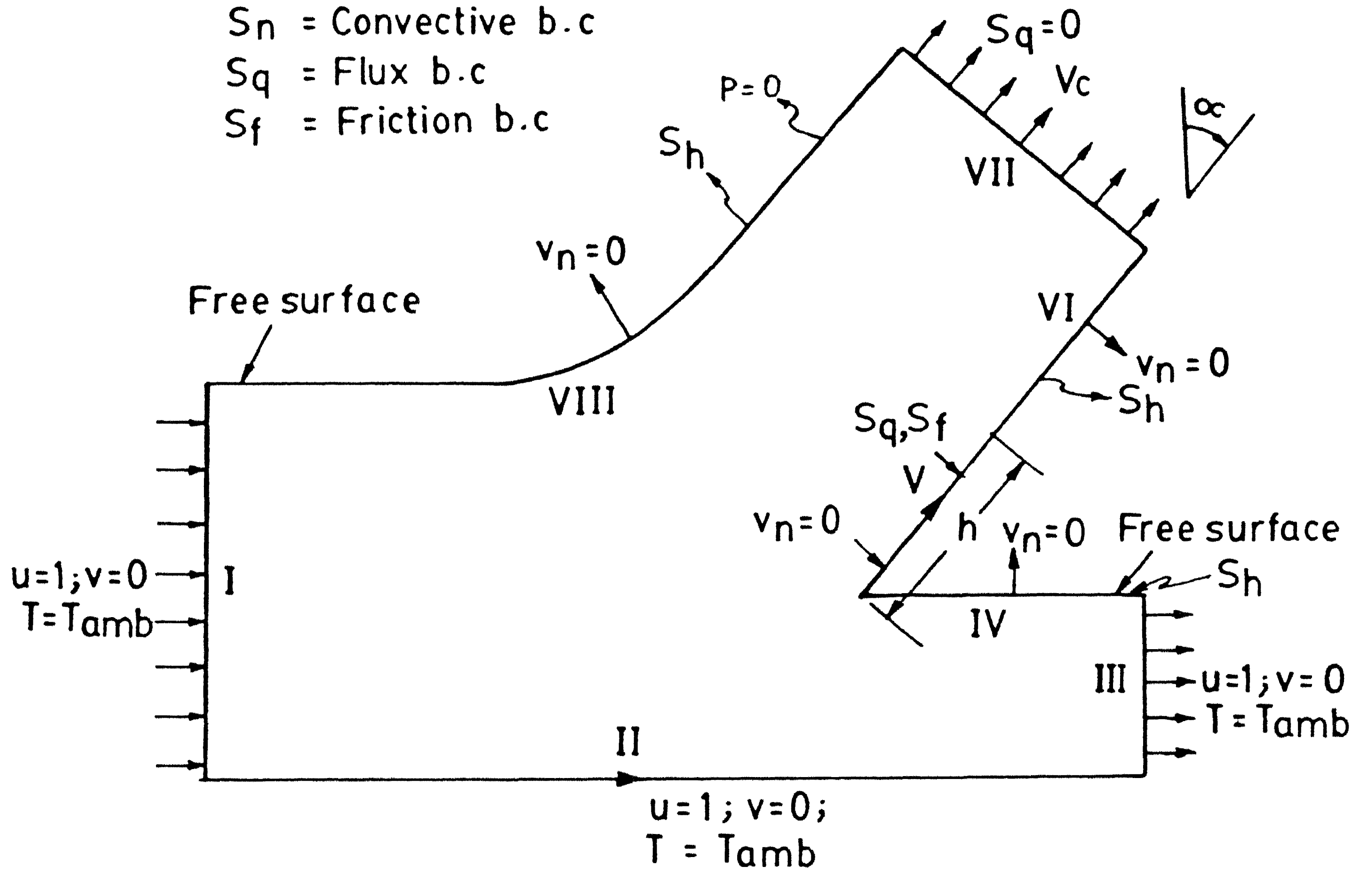


FIG. 2.3 PROBLEM REGIONS, SHOWING THE BOUNDARY CONDITIONS

from the actual region of plastic flow and hence have prescribed velocity and temperature boundary conditions. The velocity components in the x and y directions are taken as unity and zero, respectively, while the temperature can be safely prescribed as the room-temperature.

The fourth surface, which is the already machined portion of the work piece may be assumed to be straight. Here the surface may be taken to be free of any boundary traction and the normal velocity may be taken to be zero. Thus,

$$\underset{\sim}{F}_s = 0 \qquad (2.34a)$$

and,

$$\underset{\sim}{V}_n = o \qquad (2.34b)$$

where $\underset{\sim}{F}_s$ is the traction force and $\underset{\sim}{V}_n$ is the normal velocity.

Also, since the surface is in contact with the atmosphere, the convective boundary condition is applied for heat transfer in the form

$$-k \frac{\partial T}{\partial n} = h\ (T - T_\infty) \qquad (2.34c)$$

where n is the coordinate normal to the surface.

The fifth surface whose length is assumed as an input data (contact length between the chip and the tool), is one of the most difficult surfaces to be handled since a number of interesting events occur on this surface. Since the chip is in perfect contact with the tool over this surface, the normal velocity on this surface can be prescribed to be zero. Thus,

$$\underset{\sim}{V}_n = 0 \qquad (2.35a)$$

A point to be highlighted for this surface is that unlike all the other segments, this experiences boundary tractions arising from the frictional and compressive stresses due to direct contact with the tool rake-face. However, the normal traction need not be specified since $\underset{\sim}{V}_n$ has been prescribed as zero. As regards frictional stress, it is assumed that this stress is given by the product of the local yield stress and a constant friction factor. Thus,

$$\tau_f = m \frac{\bar{\sigma}}{\sqrt{3}} \qquad (2.35b)$$

$$\Rightarrow \text{from eq. 2.11c} \quad \tau_f = m \mu \dot{\bar{e}} \qquad (2.35c)$$

Assumption of a constant friction factor for the entire contact length is questionable, as discussed subsequently. However, for want of a better estimate of variation of m along the contact patch, a constant value has been assumed.

Regarding the heat transfer conditions, the heat generated by friction at the surface has to be accounted for. This is a very important condition, as it leads to the occurrence of a high temperature region over the contact patch. In the present study, it is assumed that the heat generated due to frictional rubbing is shared between the chip and the tool in the ratio of the respective thermal conductivities. Thus,

$$-k\frac{\partial T}{\partial n} = \frac{k}{k_t}\dot{Q}_f \qquad (2.35d)$$

where k and k_t are the thermal conductivities of the chip and the tool and $\dot{Q}_f$ is the rate of heat generation per unit area due to friction. From basic mechanics the frictionl power dissipation is given by the product of shear stress and velocity. Mathematically,

$$\dot{Q}_f = \tau_f V_T \qquad (2.35e)$$

where τ_f = Effective frictional shear stress $(= m\frac{\bar{\sigma}}{\sqrt{3}}$ or $m\mu\dot{\bar{\varepsilon}})$

V_T = Tangential velocity of slip

$\dot{Q}_f$ = Rate of Frictional Heat Generation.

The tangential slip velocity V_T can, in turn, be evaluated as :

$$V_T = v\ \cos\alpha + u\ \sin\alpha \tag{2.35f}$$

where α = Rake Angle of Tool.

Combining equations 2.29c, 2.29d, and 2.29e the final form of heat transfer equation for the fifth surface is :

$$-k\frac{\partial T}{\partial n} = \frac{k}{k_t}\left\{\frac{m\bar{\sigma}}{\sqrt{3}}\ (v\cos\alpha + u\sin\alpha)\right\} \tag{2.35g}$$

For the sixth and eight surfaces, the conditions of zero normal velocity, zero surface traction and convective heat loss to the atmosphere are applicable. Therefore, on these surfaces,

$$\underset{\sim}{V}_n = 0 \tag{2.36a}$$

$$\underset{\sim}{F}_s = 0 \tag{2.36b}$$

and,
$$-k\frac{\partial T}{\partial n} = h_{surf}\ (T-T_{amb}) \tag{2.36c}$$

For the seventh surface, which is on the chip exit side, the following conditions are applicable. From a mass balance between the uncut and the cut material, the chip velocity V_c can be estimated. Since the chip material is assumed to become rigid once again far away from the tool tip, the velocity can be taken to be equal to V_c on this boundary. The velocity components are then given by

$$u = V_c \sin\alpha \tag{2.37a}$$

$$v = V_c \cos\alpha \tag{2.37b}$$

where

$$V_c = V \frac{t}{t_c} \tag{2.37c}$$

For heat transfer, there is no suitable boundary condition. It is assumed that the temperature of the chip becomes invariant in the chip-flow direction. Thus,

$$\frac{\partial T}{\partial n} = 0 \tag{2.37d}$$

NON-DIMENSIONALIZATION

The non-dimensional form of the boundary conditions are:

For convective heat-loss on boundaries IV, VI and VIII,

$$\Rightarrow -\frac{\partial T^*}{\partial n^*} = h^* \ (T^* - T_f^*) \tag{2.38a}$$

$$\text{where } h^* = \frac{h}{(k/t)} \quad \text{and} \quad T_f^* = \frac{T_f}{T_{ref}} \tag{2.38b}$$

On the chip-tool interface (surface V)

$$-\frac{\partial T^*}{\partial n^*} = \frac{k}{k_t} \cdot Pe \ \tau^* \ V_t^* \tag{2.39}$$

On surfaces I, II and III,

$$T^* = T_f^* \tag{2.40}$$

At the far end of the chip (surface VII),

$$\frac{\partial T^*}{\partial n^*} = 0 \tag{2.41}$$

The dimensionless velocity conditions are :

$$u^* = 1, \; v^* = 0 \quad \text{on surfaces I, II and III} \tag{2.42}$$

$$v_n^* = 0 \quad \text{for surfaces IV, V, VI and VIII} \tag{2.43}$$

The chip velocity V_c can also be non-dimensionalized as

$$V_c^* = \frac{V_c}{V} \tag{2.44a}$$

resulting in

$$\left.\begin{aligned} u^* &= V_c^* \cos\alpha \\ v^* &= V_c^* \sin\alpha \end{aligned}\right\} \text{For surface VII} \tag{2.44b}$$

The frictional stress condition on surface V becomes :

$$\tau_f^* = \frac{1 + B^* (\dot{e}^*)^{1/n}}{\sqrt{3}} \tag{2.45}$$

The dimensionless governing equations and boundary conditions formulated above have been solved by the application of the Finite Element Method. The details of the solution procedure are described in the next chapter.

CHAPTER 3

FINITE ELEMENT ANALYSIS

In many engineering problems, it is not always possible to obtain a closed form exact solution. Naturally, recourse needs to be taken towards approximate solutions in such cases. Fortunately, with the advent of digital computers, the effective application of approximate numerical techniques in almost all branches of engineering has widened the scope of software solutions by leaps and bounds. The finite Element Method adopted for the present study is one such powerful technique for obtaining numerical solutions to difficult problems.

3.1 WEIGHTED RESIDUAL METHOD

The differential equations corresponding to an engineering problem are required to be satisfied at every point in the solution domain. However the numerical solution for the problems may not satisfy the governing equations exactly. Instead, it may satisfy the equations approximately, leaving a small non-zero residue at most of the locations in the solution domain.

Denoting the exact and approximate solutions by Φ and Φ^*, the substitution of these solutions into the governing differential equations yields

$$\left.\begin{array}{llll} L(\Phi) & = & 0 & \text{Exact Solution} \\ L(\Phi^*) & = & R & \text{Approx. Solution} \end{array}\right\} \qquad (3.1)$$

$$
\left.\begin{array}{llll} L(\Phi) & = & 0 & \text{Exact Solution} \\ L(\Phi^*) & = & R & \text{Approx. Solution} \end{array}\right\} \quad (3.1)
$$

where L is the differential operator of the problem .

From the above expressions, it is obvious that the smaller the value of residue, the closer the approximate solution is to the exact one. An alternate way of looking at it is that the larger number of points at which the residue is small, the closer are the solutions Φ and Φ^*. Since, limitations exist on the computational time and effort, it may not be possible to increase this number beyond a certain value. Therefore, methods have been developed by which the residue is minimized in an average (integrated) sense over the solution domain. In order to pin-down the residue to small values at chosen number of points in the domain, approximate weighting functions are multiplied during the minimization of the residue in the whole solution domain.

The above discussed objectives are achieved by setting,

$$
\int_D W_i \, R \, dv = 0 \qquad (3.2a)
$$

for $i = 1,..n$ (selected nodes)

where W_i are chosen weighting functions and D is the solution domain.

Substituting eq. 3.1 into eq. 3.2a, it gives

$$\int_D W_i \; L(\Phi^*) \; dv = 0 \qquad (3.2b)$$

$$\text{for } i = 1,..n$$

Equations (3.2b) form the basis of all Weighted Residue methods, for solving differential equations. The manner in which the weighting functions are selected and the approximate solution Φ^* is defined D, leads to many different weighted residual methods such as Ritz Method, Galerkin Method, Collocation Method and the method of Least Squares. The Garlerkin's method has a general applicability and hence been adopted here for the Finite Element solution procedure.

3.2 FINITE ELEMENT PROCEDURE

The finite element procedure as the name indicates, implies that the solution domain is divided into many finite sub-domains (or elements). This discretization is advantageous particularly for complex domains since the focus of attention now is reduced to that of an element of chosen shape. Considering the element as a building block, complicated domain shapes can be represented, even by the use of elements of a particular shape. Thus the analysis is very much generalized and computation becomes easier, natural question which then arises is how these elements are related or clubbed together. The answer lies in the fact that since the

variables vary continuously across the elements, a group of elements could be assembled to form a larger domain.

The sequence of procedures in any FEM solution strategy could be outlined as follows.

1. Discretization of domain into small elements.
2. Derivation of elemental properties.
3. Grouping the elements into a global assembly.
4. Incorporation of Boundary Conditions.
5. Solving the resulting global Matrix Equations to obtain the field variables.
6. Any processing that can further be done.

The above mentioned steps are discussed in detail later. The advantage in splitting the domain into many elements is that the field variables can be interpolated using standard interpolation functions within each element. For this purpose, nodes are selected within each element and the unknown variable is expressed in terms of the values of field variable at the selected nodes. The interpolating functions for an element are also called as Shape Functions. Every node has a shape function whose value is unity at that particular node and zero at all the other nodes of that element. This is a common property of all the shape functions, although in their form they may differ from each other.

Mathematically, the value of the field variable within an

element can be expressed as

$$\Phi^{*} = \sum_{i=1}^{n} N_i \Phi_i \qquad \text{where } \Phi_i \text{ are nodal values} \tag{3.3}$$

Since such representations are possible for all the elements in the domain, it can be said that the field variable is known in the entire domain. Using standardized shape functions, elemental shapes and nodal locations, the elemental properties are easily described. These, in turn, aid the computation of the solution on a computer.

3.3 TYPES OF FLOW FORMULATION

The nature of a flow problem and the flow quantitites of interest (say velocities, pressure, vorticity etc.) influences the solution approach. Based on these considerations different types of flow formulations have been developed which are explained briefly here.

The Stream-Function approach, first proposed by OSLON [19], reduces the number of variables in the flow problem to a single variable. This new variable stream function Ψ is defined in such a way that it automatically satisfies the mass balance equation. Also the Momentum Equations are reduced in terms of the stream function. But this process results in a higher-order differential equation (fourth order) and demands higher-order interpolations within the elements which is disadvantageous in some cases. More

importantly, to obtain the velocities results have to be further processed. In the present problem, since velocities are directly used in the energy equation, this approach is not adopted.

In the **Stream-Function Vorticity** approach an additional variable called vorticity (represented as ω) is defined and the three equations of mass and momentum balances are reduced to two 2nd order equations in Ψ and ω. As mentioned above, the primary interest in the present problem being velocities, this approach is also not suitable.

The third method called the **Velocity-Pressure** involves the direct solution of governing equations. There are many advantages for this approach such as applicability to three-dimensional flow, easy incorporation of pressure and velocity boundary conditions and so on.Also zeroth order continuity being sufficient for the elemental interpolation functions, this approach requires less computational time than the other approaches. Often, to eliminate some of the numerical difficulties, the pressure variable is elimінted by taking mass balance as a constraint and introduc into the momentum equation. This is done by what is called as the penality function procedure. But this

needs that the pressure boundary conditions be modified accordingly and hence is believed to be unsuitable for the present problem.

3.4 APPLICATION OF FEM

A. DOMAIN DISCRETIZATION

Discretization of the domain involves division of the solution domain into small elements. For plane two-dimensional problems, these are n-sided polygons and hence many types of elements are possible, like triangular,quadrilateral etc. The domain discretization, seemingly simple, can affect the accuracy of the solution considerably, if done improperly. This is particularly so when the domain is of a complex shape. Isoparametric elements (so-called because the field variable and the spatial coordinates can be defined with the help of the same interpolation functions)offer many advantages and hence employed in the present study (shown in Fig. 3.1). Eight-noded linear quadrilateral elements have been used, as shown in Fig. 3.2, in creating the finite element mesh.

B. DERIVATION OF FINITE ELEMENT EQUATIONS

The finite element equations for plastic flow of metal, are better derived from the basic vector equation (eq. 2.13) so that the handling of the traction boundary conditions at the chip-tool interface could be easily implemented.

FLOW EQUATIONS

The vector Momentum Equation eq. 2.14 is given by

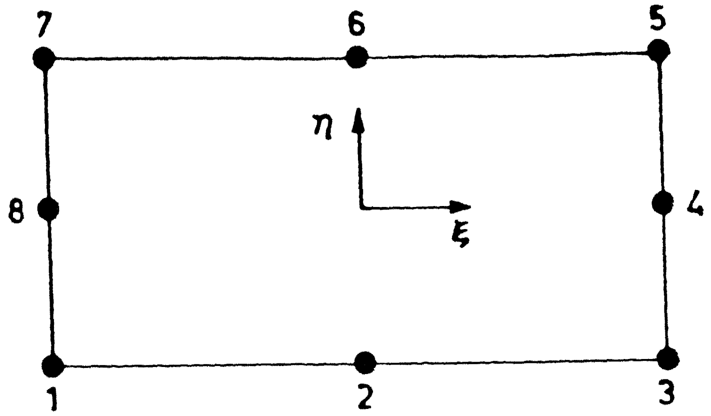


Fig. 3.1. Eight Noded Element Showing the Local Co-ordinate System.

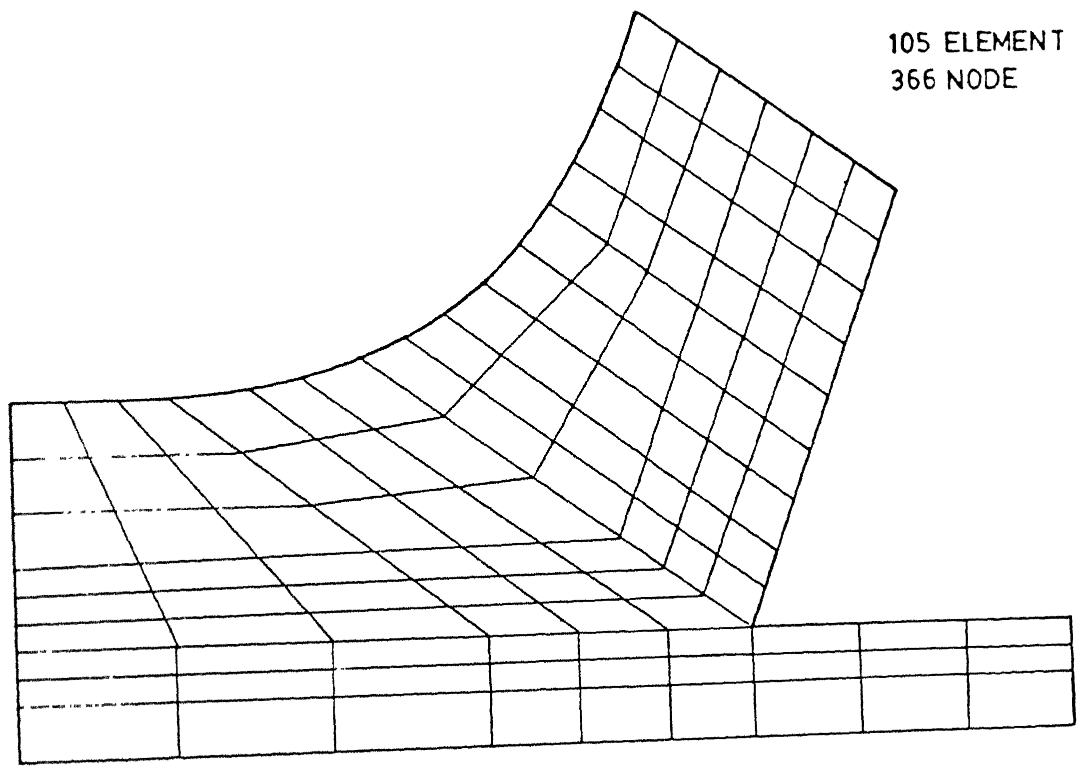


FIG. 3-2 FINITE ELEMENT MESH FOR RAKE ANGLE 20°

$$\rho\,(\underset{\sim}{V}\,.\,\underset{\sim}{\nabla}\,\underset{\sim}{V})\;-\;\underset{\sim}{\nabla}\,.\,\underset{\approx}{\sigma}\;-\;\underset{\sim}{f}\;=\;[0]$$

By Galerkin technique, the vector residue equation for two-dimensional cutting can be formed as

$$\iint N_i \left\{ \rho\underset{\sim}{V}.\,\underset{\sim}{\nabla}\,\underset{\sim}{V} - \underset{\sim}{\nabla}.\underset{\approx}{\sigma} - \underset{\sim}{f} \right\} dx\,dy \;=\; [0] \qquad (3.4a)$$

$$=> \iint N_i\,(\rho\underset{\sim}{V}\,.\,\underset{\sim}{\nabla}\,\underset{\sim}{V})dxdy - \iint N_i\,(\underset{\sim}{\nabla}.\underset{\approx}{\sigma})\,dxdy - \iint N_i\,\underset{\sim}{f}\,dxdy \;=\; [0]$$

(3.4b)

The term $\underset{\sim}{f}$ in eq. 3.10 can be included in the pressure by redefining pressure as

$$\underset{\sim}{\nabla}\,p' = \underset{\sim}{\nabla}p + \underset{\sim}{f} \qquad (3.5)$$

The term $\iint N_i\,(\underset{\sim}{\nabla}.\underset{\approx}{\sigma})$ dxdy can be written by the Weak Formulation as

$$\iint N_i\,(\underset{\sim}{\nabla}.\underset{\approx}{\sigma})\,dxdy \;=\; \iint_D \left[\underset{\sim}{\nabla}.(N_i\,\underset{\approx}{\sigma}) - \underset{\sim}{\nabla}\,N_i\,.\underset{\approx}{\sigma} \right] dxdy$$

(3.6a)

Expanding the first term in parenthesis by divergence theorem, the above equation 3.6a reduces to

$$\iint N_i(\underset{\sim}{\nabla}.\underset{\approx}{\sigma})dxdy = \oint_B N_i\underset{\sim}{n}\,.\,\underset{\approx}{\sigma}\,dl - \iint \underset{\sim}{\nabla}\,N_i\,.\,\underset{\approx}{\sigma}\,dxdy \qquad (3.6b)$$

where in the first term gives the contributions due to the specified boundary conditions. The original momentum residue equation 3.4b can now be rewritten as

$$\iint N_i \ \rho \ (\underset{\sim}{V} \cdot \underset{\sim}{\nabla}) \ \underset{\sim}{V} \ dxdy + \iint \underset{\sim}{\nabla} N_i \ \cdot \underset{\approx}{\sigma} \ dxdy$$

$$= \oint N_i \ \underset{\sim}{n} \ \cdot \underset{\approx}{\sigma} \ dl \qquad (3.7)$$

Expanding $\underset{\approx}{\sigma}$ into its components, the LHS term $\underset{\sim}{\nabla} N_i . \underset{\approx}{\sigma}$ can be written as

$$\underset{\sim}{\nabla} N_i . \underset{\approx}{\sigma} = \left\{ \frac{\partial N_i}{\partial x} \sigma_{xx} + \frac{\partial N_i}{\partial y} \sigma_{yx} \right\} \hat{i} + \left\{ \frac{\partial N_i}{\partial x} \sigma_{xy} + \frac{\partial N_i}{\partial y} \sigma_{yy} \right\} \hat{j} \qquad (3.8)$$

X-Momentum

Putting the above term of eq. 3.8 into the momentum equation 3.7, the x and y component equations can be derived as

$$\iint \rho N_i (\bar{u} \frac{\partial u}{\partial x} + \bar{v} \frac{\partial u}{\partial y}) \ dxdy$$

$$+ \iint \left[\left\{ \frac{\partial N_i}{\partial x} (- p + 2\mu \frac{\partial u}{\partial x}) \right\} + \frac{\partial N_i}{\partial y} \left\{ \mu (\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}) \right\} \right] dxdy$$

$$= \oint N_i \ (\underset{\sim}{n} . \underset{\approx}{\sigma} . \hat{i}) \ dl \qquad (3.9a)$$

and

Y-Momentum

$$\iint \rho N_i \left(\bar{u}\frac{\partial v}{\partial x} + \bar{v}\frac{\partial v}{\partial y}\right) dxdy$$

$$+ \iint \left\{\frac{\partial N_i}{\partial x}\left[\mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right] + \frac{\partial N_i}{\partial y}\left[-p + 2\mu\frac{\partial v}{\partial y}\right]\right\} dxdy$$

$$= \oint N_i (\underset{\sim}{n} . \underset{\approx}{\sigma} . \hat{j})\, dl. \qquad (3.9b)$$

In the above equations, the inertial terms have n quasi-linearized by using the velocities corresponding to prev s iteration (or initial guess) values during the iterative solution of the problem. For this reason, these quantities have been denoted as $\bar{u}$ and $\bar{v}$.

The field variables have been interpolated within each element as shown below

$$\left.\begin{aligned} u &= \sum_{i=1}^{n} N_i\, u_i \qquad & v &= \sum_{i=1}^{n} N_i\, v_i \\ p &= \sum_{l=1}^{m} N_l\, p_l \qquad & T &= \sum_{i=1}^{n} N_i\, T_i \end{aligned}\right\} \qquad (3.10a)$$

Similarly, the Spatial Coordinates of the boundaries of the isoparametric element are defined as

$$x = \sum_{i=1}^{n} N_i x_i \quad ; \quad y = \sum_{i=1}^{n} N_i y_i \qquad (3.10b)$$

where $n = 1,8$

$m = 1,4$

It may be noted that pressure is defined only at corner nodes of the element and hence needs only a linear interpolating function.

All the other variables (u, v and T) have been interpolated using quadratic shape functions. The lower order interpolation for pressure has been found necessary to enhance numerical stability [10].

For easy computation, the interpolation function N_i are defined in terms of local coordinates as

Corner Nodes

$$N_i = \frac{1}{4} (1 + \xi_i \xi) (\xi_i \xi + \eta_i \eta - 1) (1 + \eta_i \eta) \qquad (3.11a)$$

Mid-side Nodes

$$\left.\begin{aligned} N_i &= \frac{1}{2} (1 - \xi^2) (1 + \eta_i \eta), \xi_i = 0 \\ N_i &= \frac{1}{2} (1 + \xi_i \xi) (1 - \eta^2), \eta_i = 0 \end{aligned}\right\} \qquad (3.11b)$$

Substituting the expressions for the interpolation of u, v and p into the momentum equations 3.9a and 3.9b gives the final non-dimensional form of x and y momentum equations as

X-Momentum

$$\left[\iint\left\{\frac{N_i}{\sigma}\left(N_k u_k \frac{\partial N_j}{\partial x} + N_k v_k \frac{\partial N_j}{\partial y}\right) + 2\mu \frac{\partial N_j}{\partial x}\frac{\partial N_j}{\partial x} + \mu \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y}\right\} dxdy\right][u_j]$$

$$+ \left[\iint\left\{\mu\left(\frac{\partial N_i}{\partial y} + \frac{\partial N_j}{\partial x}\right)\right\} dxdy\right][v_j]$$

$$- \left[\iint\left(\frac{\partial N_i}{\partial x} M_j\right) dxdy\right][p_j] = \left[\oint N_i \, t_x \, dl\right] \qquad (3.12a)$$

Y-Momentum

$$\left[\iint\left\{\mu\left(\frac{\partial N_i}{\partial x} \cdot \frac{\partial N_j}{\partial y}\right)\right\} dxdy\right][u_j] \; +$$

$$\left[\iint\left\{\frac{N_i}{\sigma}\left(N_k u_k \frac{\partial N_j}{\partial x} + N_k v_k \frac{\partial N_j}{\partial y}\right) + \mu \frac{\partial N_j}{\partial x}\frac{\partial N_j}{\partial x} + 2\mu \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y}\right\} dxdy\right][v_j]$$

$$- \left[\iint \left(\frac{\partial N_i}{\partial y} M_j \right) dxdy \right] [p_j] = \left[\oint N_i \; t_y \; dl \right] \qquad (3.12b)$$

where t_x and t_y are the traction components per unit area, given by

$$t_x = \underset{\sim}{n} . \underset{\approx}{\sigma} . \hat{i} \qquad (3.13a)$$

and $$t_y = \underset{\sim}{n} . \underset{\approx}{\sigma} . \hat{j} \qquad (3.13b)$$

The residue minimization principle for the mass balance (material incompressibility) equation takes the form

$$\iint_D M_i \left\{ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right\} dxdy = 0 \qquad (3.14)$$

The weighting functions for the mass balance equation have been chosen to be the linear shape functions used to interpolate pressure within an element. This is done in order to get a pressure field that is comparable with the incompressibility condition. Substituting the interpolation expression for the velocity variable (eq. 3.10a), the final form of the mass balance

equation is

$$\left[\iint M_i \frac{\partial N_j}{\partial x} dxdy \right] [u_j]$$

$$+ \left[\iint M_i \frac{\partial N_j}{\partial x} dxdy \right] [v_j] = 0 \qquad (3.15)$$

ENERGY EQUATION

The weighted residue form of the energy equation (eq. 2.24) is

$$\iint N_i \left\{ \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right\} dxdy + \iint N_i \, Pe \frac{\bar{\sigma}}{\sqrt{3}} \dot{\bar{e}} \, dxdy$$

$$- \iint N_i \left\{ u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} \right\} dxdy = 0 \qquad (3.16a)$$

Since by Green's theorem

$$\iint N_i \left\{ \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right\} = \oint N_i \frac{\partial T}{\partial n} dl$$

$$- \iint \left(\frac{\partial N_i}{\partial x} \frac{\partial T}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial T}{\partial y} \right) dxdy \qquad (3.16b)$$

expanding the temperatire within an element as $T = \Sigma N_j T_j$, the above equation reduces to

$$\left[\iint \left\{ \left(\frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right) \right. \right.$$

$$\left. \left. + N_i \; Pe \left[N_k \bar{u}_k \frac{\partial N_j}{\partial x} + N_k \bar{v}_k \frac{\partial N_j}{\partial y} \right] \right\} dxdy \right] [T_j]$$

$$= \iint N_i \; Pe \frac{\bar{\sigma}}{\sqrt{3}} \; \dot{\bar{e}} \; dxdy + \oint N_i \frac{\partial T}{\partial n} \, dl \qquad (3.17)$$

where the first term on RHS is the heat generation contribution and the second term is from the heat transfer boundary conditions.

The weighted residue equations for momentum, mass and energy balances, lead to matrix equations in terms of the nodal variables of velocity, pressure and temperature. The contributions to the matrix equations from each element can be evaluated by applying the momentum and energy equations at all eight-nodes of an element and the mass balance equations only at the four corner nodes of the element. Thus, for each element, the coefficient matrix contribution is a (28x28) matrix, while that for the right hand side is a vector of size (28x1).

After computation of the element matrices, the boundary conditions are incorporated which is explained in the next section. Later the elemental matrices and vectors are assembled into global matrices, vectors and the resulting equations are solved iteratively.

C. BOUNDARY CONDITIONS

For the present problem, the domain as well as the boundary conditions are complex. To facilitate numerical computation, certain assumptions had to be made with regard to the shape of the chip free-surface, the contact length, the friction factor on the rake-face etc. In this section, the implementation of the boundary conditions on all the eight boundary surfaces through the evaluation of the appropriate boundary integrals or by the prescription of the known values of velocity, pressure etc. is explained.

Surfaces I, II and III have prescribed boundary conditions for the variables u, v and T. These conditions are implemented by incorporating them as nodal equations for the corresponding variables, in the final matrix equations.

The normal velocity ($\underset{\sim}{V_n} = 0$) is applied on surfaces IV, V, VI & VIII, since there is no flow in a direction normal to these surfaces. This condition is satisfied by expressing the normal

velocity in terms of the velocity components u and v and the direction cosines of the normal at that node. Thus, normal velocity at the surface requires

$$\underset{\sim}{V}_n = u \, \mathrm{Cos}\, \theta \, \hat{i} - v \, \mathrm{Sin}\, \theta \, \hat{j} = 0 \qquad (3.18a)$$

where θ is the angle made by the normal with the x-axis and $\hat{i}$, $\hat{j}$ are unit vectors in x and y directions respectively.

Again, expanding the terms u and v as in eq. 3.10a, in terms of the nodal velocity values and interpolating functions., the above equation can be rewritten as

$$\underset{\sim}{V}_n = \sum_{j=1}^{8} N_j \, u_j \, \mathrm{Cos}\, \theta \, \hat{i} - \sum_{j=1}^{8} N_j \, v_j \, \mathrm{Sin}\, \theta \, \hat{j} = 0 \quad (3.18b)$$

where u_j, v_j are the nodal velocities of the element which lies on the concerned boundary.

The angle made by the surface normal with the x-direction can be given as input or can be evaluated from the element shape (specified by nodal coordinates) using the iso-parametric nature of the element.

On surface VII, which is the far end of the chip, the velocity is equal to chip velocity V_c. The velocity components u and v on this boundary are given by

$$\left.\begin{aligned} u &= V_c \operatorname{Sin} \alpha \\ v &= V_c \operatorname{Cos} \alpha \end{aligned}\right\} \qquad (3.19)$$

where α = rake angle of the tool.

On surface V, in addition to zero normal velocity, the traction condition due to interface friction is to be implemented. As shown by equations 3.12a and 3.12b, the traction contributions to the x and y momentum equations are respectively

X-Momentum

$$\int_V N_i \, t_x \, dl \qquad \text{and} \qquad (3.20a)$$

Y-Momentum

$$\int_V N_i \, t_y \, dl \qquad (3.20b)$$

where t_x and t_y are tractions per unit area, in the x and y directions. On surface V, the shear stress τ_f can be evaluated (as given by eq. 2.35c in terms of the friction factor, viscosity and the current strain-rate. Thus, in terms of τ_f, t_x and t_y reduce to the form

$$t_x = \tau_f \sin\alpha \qquad (3.21a)$$

and $$t_y = \tau_f \cos\alpha \qquad (3.21b)$$

Surfaces IV, VI and VIII are free as no traction is applied on any of these. Therefore, the traction integrals for these boundaries are identically zero and only the zero normal velocity condition is applied on these surfaces.

As regards the heat transfer boundary conditions, these involve either given temperature or flux or convective conditions. Surfaces I, II and III have prescribed temperature whose implementation has already been described. On other surfaces, the relevant boundary heat flux integral term

$$\int_{\Omega} N_i \frac{\partial T}{\partial n} \, dl \qquad (3.22)$$

needs to be evaluated.

For thesurfaces IV, VI and VIII which have convective heat transfer b.c.,the overall heat-transfer coefficient is to be evaluated from forced convection correlations for air. In the present study a simplified expression of the form

$$h = 50\sqrt{V} \qquad (3.23)$$

has been chosen, where V is the cutting velocity in m/s.

The boundary integral term of eq. 3.22

$$\int_{\Omega} N_i \frac{\partial T}{\partial n} \, dl$$

can be rewritten using the dimensionless boundary condition given in 2.38a in a form

$$\int N_i \frac{\partial T}{\partial n} \, dl = - \int_{S_h} N_i \, h \, (T - T_f) \, dl \qquad (3.23a)$$

On the surfaces IV, VI and VIII expanding T from eq. 3.10a as

$$T = \sum_{j=1}^{8} N_j T_j$$

the expression for convective heat loss becomes

$$\int N_i \, h \, (T - T_f) \, dl = \left[\int_{S_h} (N_i \, h \, N_j) \, dl \right] [T_j]$$

$$- \left[\int_{S_h} (N_i \, h \, T_f) dl \right] \qquad (3.23b)$$

In the above equation, the first term on right contributes to the left hand side of the overall matrix equation while the second one contributes to the right hand side vector.

On the chip-tool interface (surface V), the condition of frictional heat generation has been taken into account. As described in Chapter 2, the heat flux entering the chip-side is calculated through the boundary integral

$$\int_V N_i \frac{\partial T}{\partial n} \, dl = - \frac{k}{k_t} \, Pe \int_V N_i \, \tau_f \, V_T \, dl \qquad (3.24a)$$

where the dimensionless shear stress itself is evaluated from eq. 2.45 and the slip velocity V_T is given by

$$V_T = v \cos \alpha + u \sin \alpha \qquad (3.24b)$$

The frictional heat flux contribution given above is added to the right hand side of the nodal equations of the temperature variables, on surface V.

D. ELEMENT ASSEMBLY

In the previous section, the elemental contributions to the left hand side coefficient matrix and the right hand side vector were discussed in detail for the eight-noded iso-parametric quadrilateral elements. These elemental contributions are assembled into a global matrix equation by adding the entries corresponding each nodal variable appropriately.

The boundary integral contributins on both left hand and right hand sides parts of the matrix equation are also incorporated as discussed in the previous section. This assembly procedure results in a matrix equation of the form

$$[A]\ [X] = [B] \tag{3.25}$$

where [A] = Coefficient matrix

[X] = Solution vector

[B] = Right hand side vector.

The problem at this stage is set for solving the above matrix equation by any standard technique available.

E. MATRIX SOLUTION TECHNIQUE

The selection of the matrix solution technique depends upon the matrix characteristics such as diagonal dominance TRID structure or pentadiagonal structure etc.

For the present problem an iterative technique is used for matrix solution based on the Frontal method. The Frontal method offers many advantages such as small use memory space and high speed of computation. It utilizes the fact that when all the contributions for a particular node and for a particular variable are over during element assembly, it could be transferred to the

disk memory. On this basis it retains in its memory only those variables which are yet to be assembled. After all the elements are assembled, the variables are solved by Gaussian elimination. Since the governing equations are highly non-linear an iteration procedure with successive under-relaxation has been employed.

3.5 PROGRAMMING

Computer Programming has been one of the challenging tasks of the present work. The challenges arose both due to the complex nature of the problem and the boundary conditions. A brief description of the programming aspects are discussed here for the benefit of interested readers.

Appendix C gives the flow chart of the program, emphasising mainly what each of the subroutines do. The Main Program calls the Subroutines of **DIMENS, DINPUT, ITERAT** which in turn call the other subroutines[Appendix D].

Subroutine **DIMENS** returns the values of the array dimensions which remain unchanged through out the program. Subroutine **DINPUT** reads and returns all of the required input parameters such as the axi-symmetric flow indicator, the number of initial and boundary conditions, body forces, tolerance, relaxation factors, coordinate information, scale factors, element connectivity data and the prescribed values of parameters like the Ambient temperatures,

material properties etc. The whole input data for the problem is non-dimensionalized within the program. Also some of the given input is checked for correction by two routines named **DIAGN1** and **DIAGN2**.

Subroutine **ITERAT** calls the **FRONTS** routine for solving the assembled matrix equation. This routine begins the problem with guessed initial values for all the nodal variables and relaxes them for the next iteration in the subroutine **TOLREL** if the solution is found not to have converged within the prescribed tolerance. This routine also calls the **WRITER** subroutine (i) to write the nodal values of the variables if the solution has converged and (ii) to write the largest relative change in the variables that has occured during a particular iteration, if the solution has not converged.

The **FRONTS** does the assembling of a group of element and finally solves for the variables by Gaussian elimination and back-substitution. It stops the execution of the program under two conditions. Firstly, when the front width prescribed is found to be too sall or if a pivot value is foud to be smaller than the set value (usually $\approx 10^{-10}$). For assembling it calls the **MATRIX** subroutine which calculates the element fluid matrix FLUMX. This fluid matrix consists of the elemental contributions of the governing equations to form the coefficient matrix. Similarly, the elemental right hand side vector is also calculated in the subroutine **MATRIX** for each iteration, the subroutine **DRIVES** is

called by **MATRIX**, which in turn calls the subroutines **SHAPE8**, **DJACOB** & **SHAPE4** to calculate the shape function and their derivatives.

The boundary conditions are implemented in the **BOUCON** which is called by **MATRIX** routine before it returns to **FRONTS** routine. This is called by identifying the surface, element and the side which are forming the boundary. The relevant boundary conditions from among the zero normal velocity, convective heat loss, the frictional heat flux and the frictional shear stress are identified appropriately and the elemental contributions are evaluated. The contributions to right hand side vector are calculated at local nodes and are assembled in the global vector both in **MATRIX** and **BOUCON** subroutines.

CHAPTER 4

RESULTS AND DISCUSSIONS

The velocity and the temperature fields in the vicinity of the tool edge during metal-cutting have been obtained by the finite element solution of coupled stress balance and heat balance equations. The work material has been taken to be purely strain-rate sensitive and obeying the Von-Mises yield criterion. A parametric study has been conducted to highlight the effects of some of the important process parameters such as Cutting Velocity, Depth of Cut and Chip Thickness Ratio. The properly values and the range of parameters used in the study are summarized in Appendix A.

From the velocity and temperature fields some of the other features of interest such as the shapes of primary and the secondary deformation zones, the location of the shear plane and the mean width of PSDZ have been obtained. The effects of cutting parameters upon the temperature variation over the chip-tool interface region, have been studied. This is very important since the highest temperatures occur in this region, causing a large tool wear. The structure of the primary and secondary deformation zones have been further analyzed by plotting the strain-rate variation in these zones. Comparisons with earlier works have been provided where available.

4.1 DEFORMATION PATTERNS AND ISOTHERMS

Figures 4.1 to 4.6 give the Deformation patterns and the corresponding Isothermal patterns for six different sets of cutting conditions. The data for the cutting parameters have been taken from [16]. In order to obtain the deformation patterns,the following procedure has been employed. The strain-rates ($\dot{e}$)at different points were first normalized with respect to the maximum strain rate occuring in the solutiòn domain. Points having the same value of normalized strain-rate were connected by smooth curves and the curves of 3%, 4% and 5% strain-rate (as compared to the maximum) were thus plotted. These constant strain-rate curves provide a clear picture of the primary and secondary deformation zones (PSDZ and SSDZ). The isotherms have been plotted by interpolating the available nodal temperatures.

In Fig. 1(a), the deformation pattern is depicted with the help of 1%, 2%, 3%, 4% and 5% strain-rate curves. The above values were used to obtain an idea of the approximate cut-off value that can be taken to demarcate the primary shear deformation zone. It can be seen that curves of 3%, 4% and 5% cut-off values are relevant in this context and hence been used for other figures (Fig. 4.2 to Fig. 4.6). Indeed it appears that the 5% curve possesses a good correspondence with experimentally observed shapes, although an appropriate cut-off value can be proposed only after a detailed comparison of the computed results and the photo-micrographs for similar cutting conditions.

The following interesting features can be noted from Fig.1(a). Firstly the constant strain-rate curves on the uncut material side are more closely placed than the cut material side (in the chip). The reason for this appears to be that theincoming material undergoes deformation due to the shearing action in the vicinity of the shear plane, caused by the large shear stresses prevailing there. Within the chip, however, the deformation is due to the curling of the chip and the compressive force of the tool on the chip. Since one of the surfaces of the chip is unrestrained, the deformation in the chip occurs over a wider region. It is to be noted, however, that the deformation near the free surface does have some dependence on the prescribed shape of this surface. The figure shows that a small region in the workpiece directly below the tool-tip is also getting deformed, because this is the place where the material is getting seperated from the workpiece. Commensurate with the high stress intensity here, the strain-rate curves are also closely spaced. The secondary deformation zone (SSDZ) is observed to be approximately triangular in shape which is usually seen in experiments too. As expected this zone extends only till the contact length for the chip with the tool. Close spacing of the strain-rate curves is seen for SSDZ also, as discussed in the other zones, due to large stresses.

For the temperature field also (in Fig. 4.1(b)), the trends seem to be realistic. The maximum temperature occurs somewhere in the region where the chip is in contact with the tool. This has

been noted by earlier researchers as well and is explained in terms of the fact that in the SSDZ further rise in temperature occurs due to frictional heating beyond the level of heating in PSDZ. For the same reason, the gradients in the secondary deformation region are the highest. The lowest temperatures in the chip occur in the middle rather than at the free surface in accordance with the deformation pattern. Bulk of the material is at room temperature not being influenced much by the heat generation. This is because of the large amount of heat removal by the chip. The temperatures in the primary zone are far less when compared to those occuring in the secondary zone. This also establishes that the frictional heat generation on the chip-tool interface is very large as compared to the heat production in PSDZ. The workpiece material below the tool-tip is also getting heated because of high deformation in this region. The occurence of very high temperature on the rake face of the tool has an important bearing upon the tool wear. The coupled analysis presented here can therefore be utilized to calculate the stresses and the temperature on the tool surface which, in turn, can be linked to tool wear.

Figures 4.1 to 4.6 also show the affects of the cutting conditions on the deformation and isothermal patterns. Particularly, Figs. 4.7 to 4.9 respectively show the effects of cutting velocity, chip thickness ratio and depth of cut on the temperatures at the chip-tool interface. The deformation regions (both primary and secondary) are wider for larger cutting

Cutting conditions			
r	v	t	m
0·757	1·65	3·0	0·637

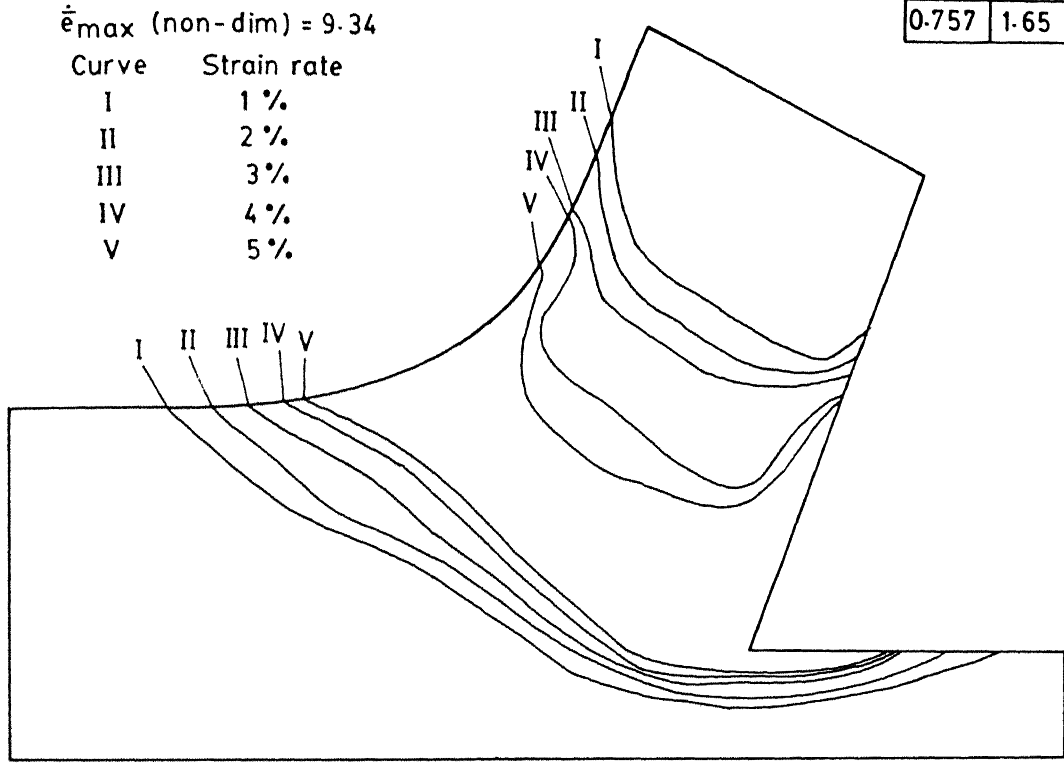


FIG. 4·1 (a) DEFORMATION PATTERN

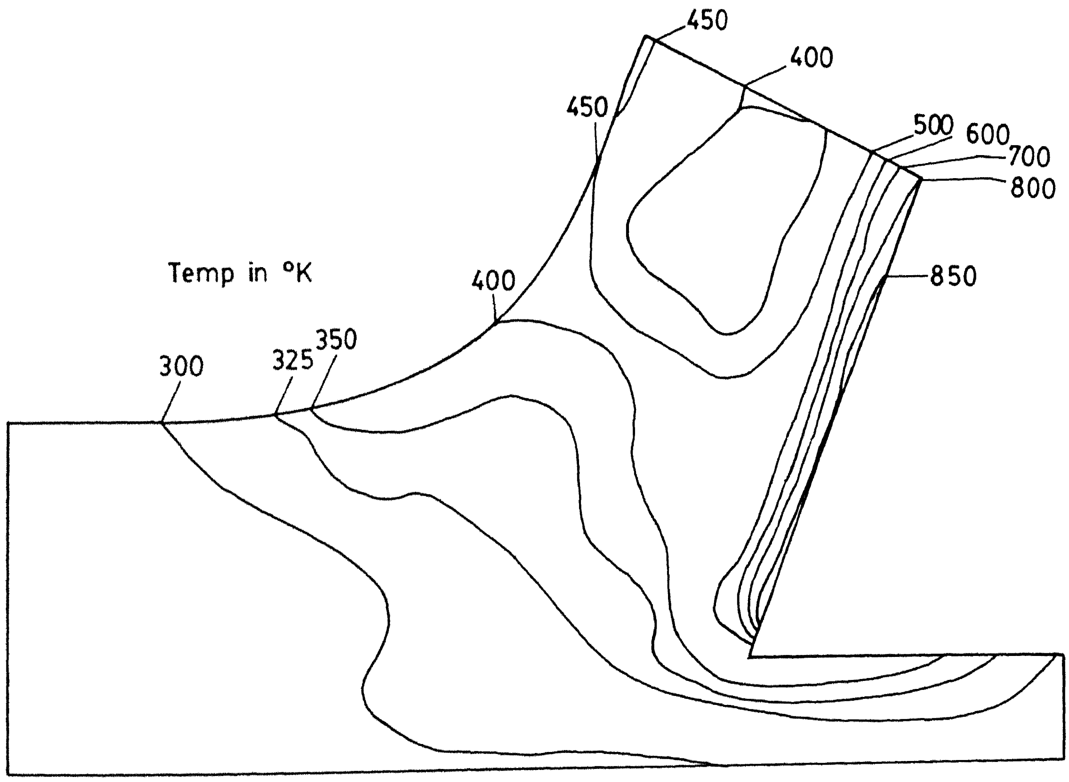


FIG. 4·1 (b) ISOTHERMAL PATTERN

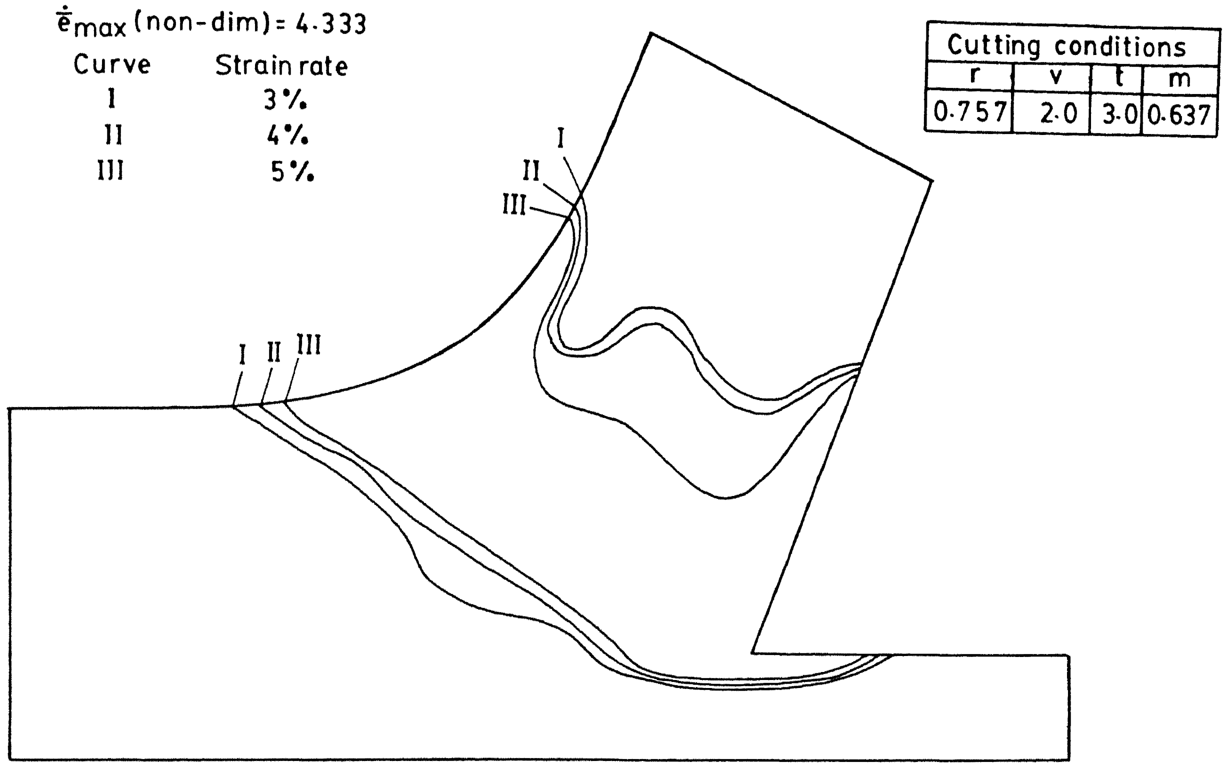


FIG 4.2(a) DEFORMATION PATTERN

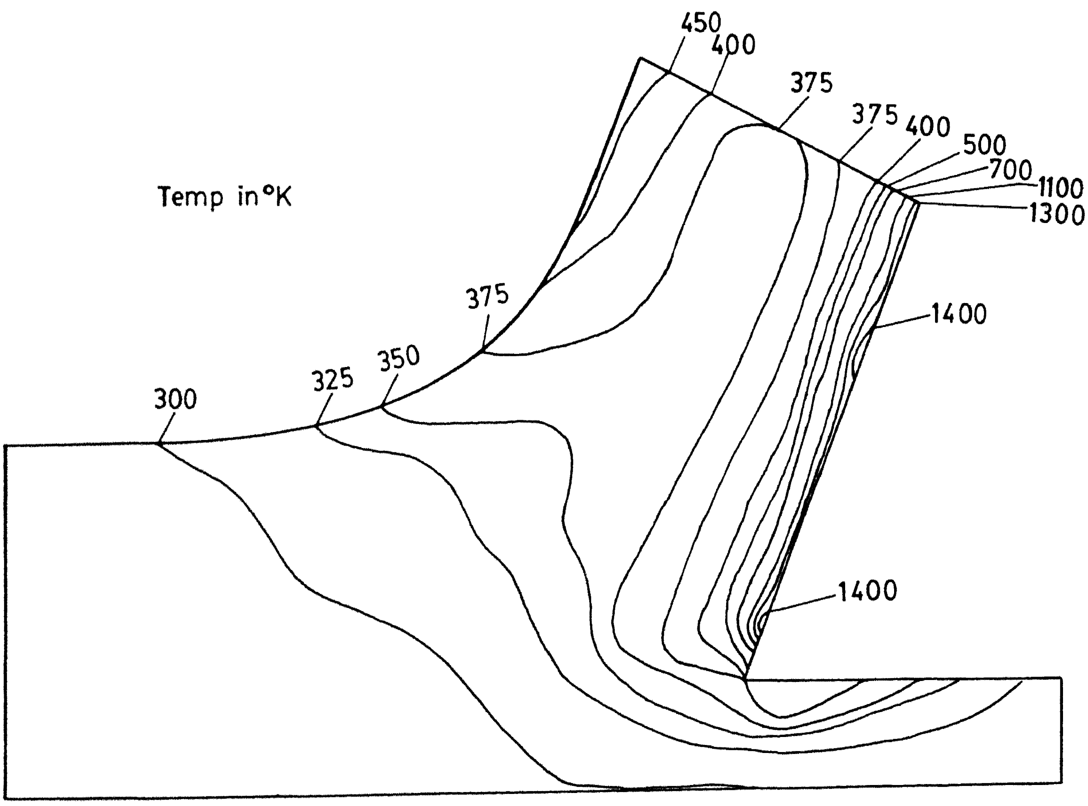


FIG. 4.2(b) ISOTHERMAL PATTERN

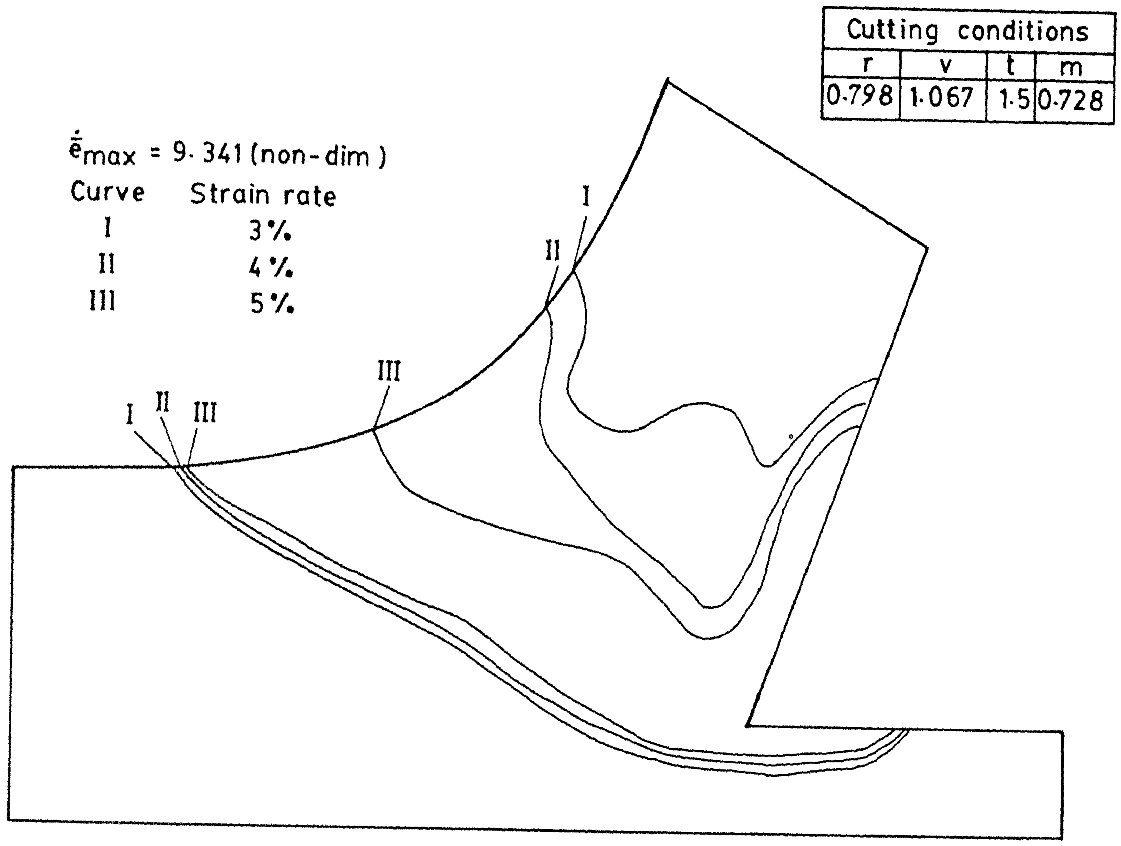


FIG. 4·3(a) DEFORMATION PATTERN

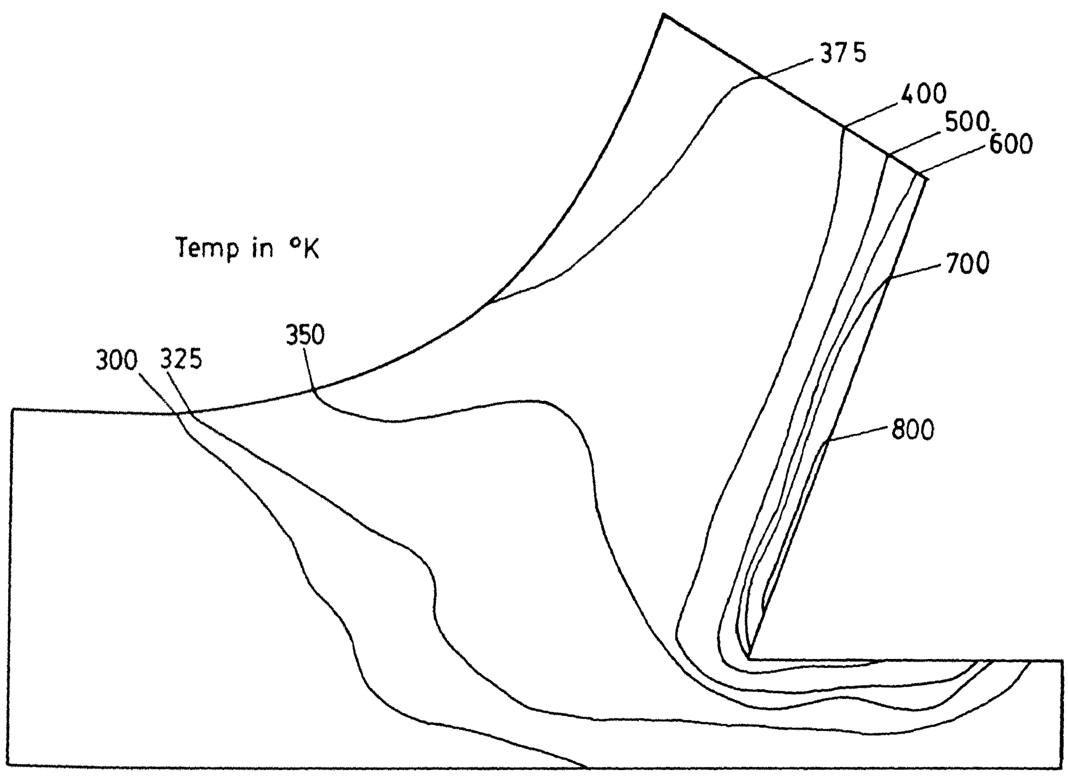


FIG. 4·3(b) ISOTHERMAL PATTERN

Cutting conditions			
r	v	t	m
0.798	1.65	1.5	0.728

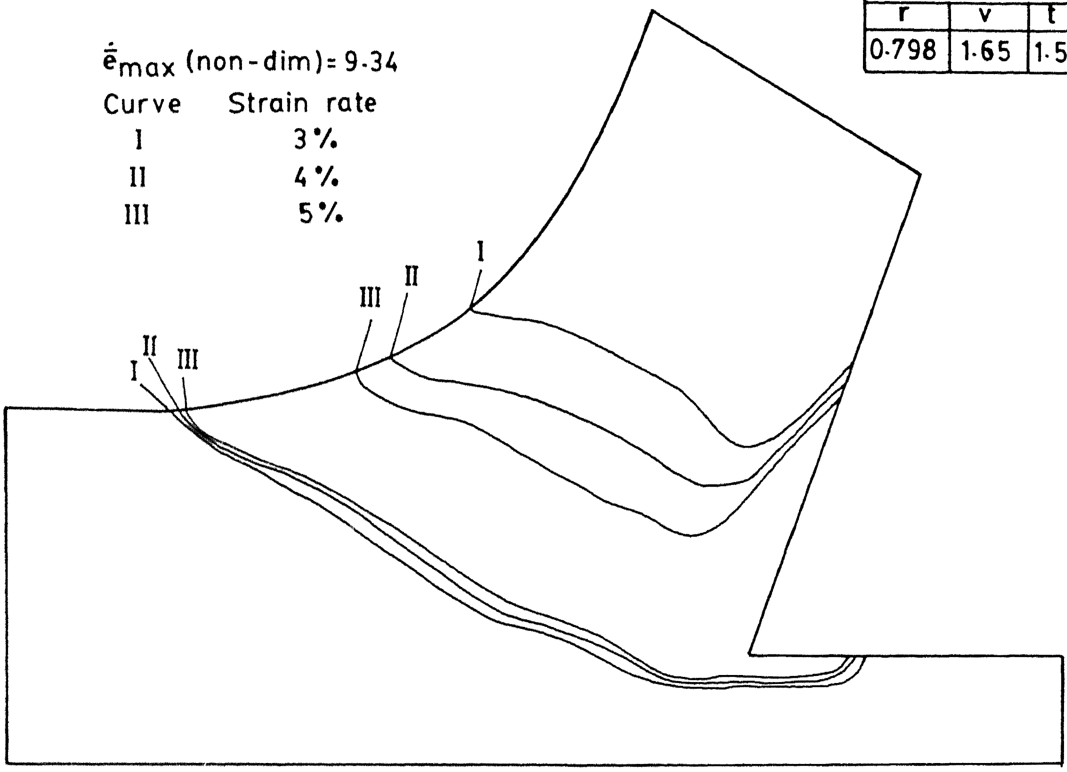


FIG. 4.4 (a) DEFORMATION PATTERN

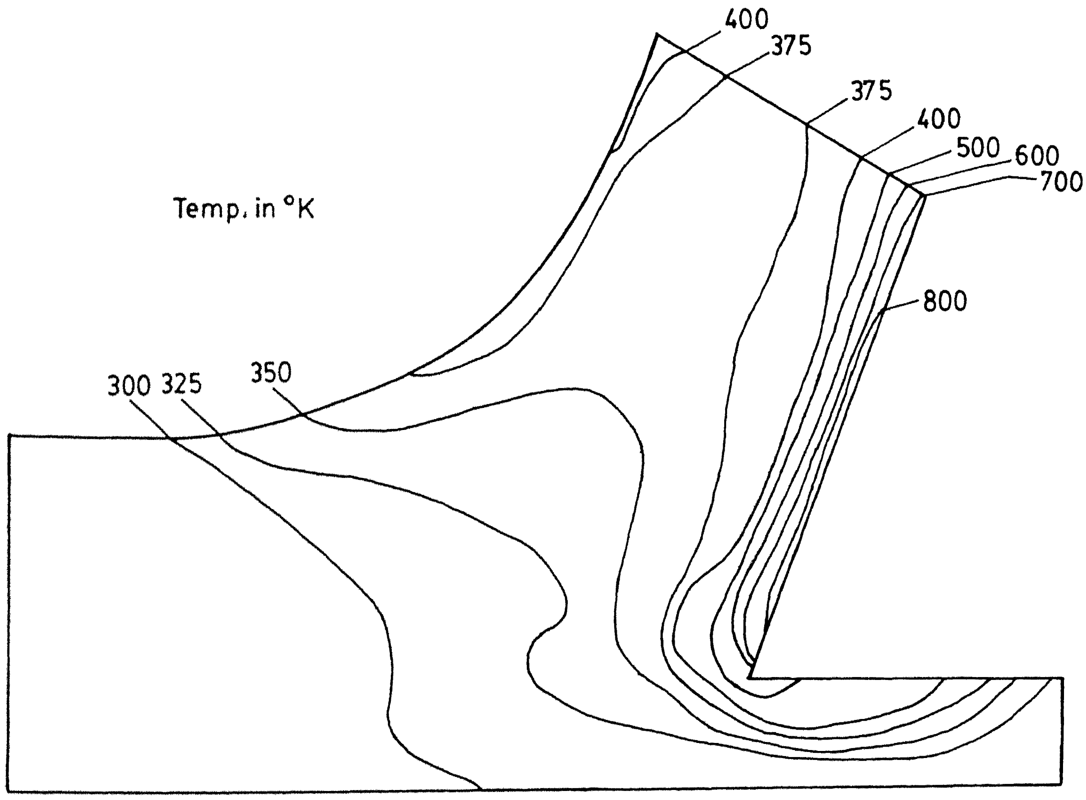


FIG. 4.4(b) ISOTHERMAL PATTERN

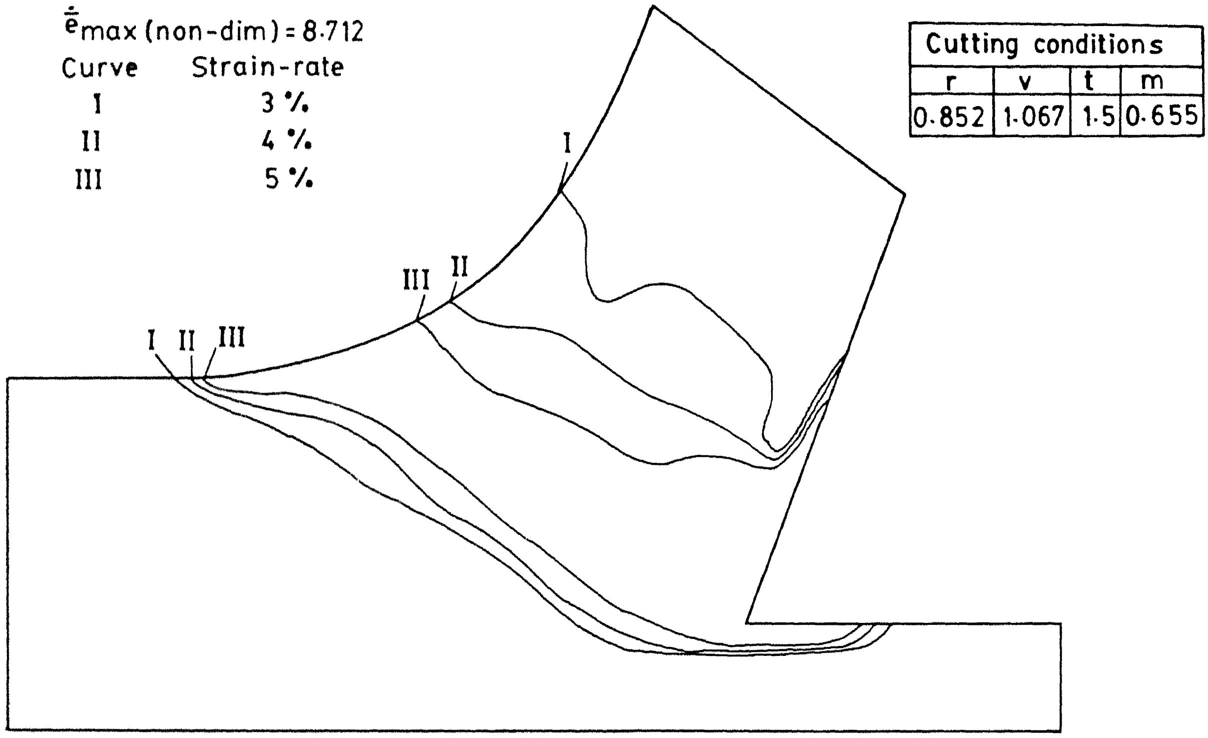


FIG. 4·5 (a) DEFORMATION PATTERN

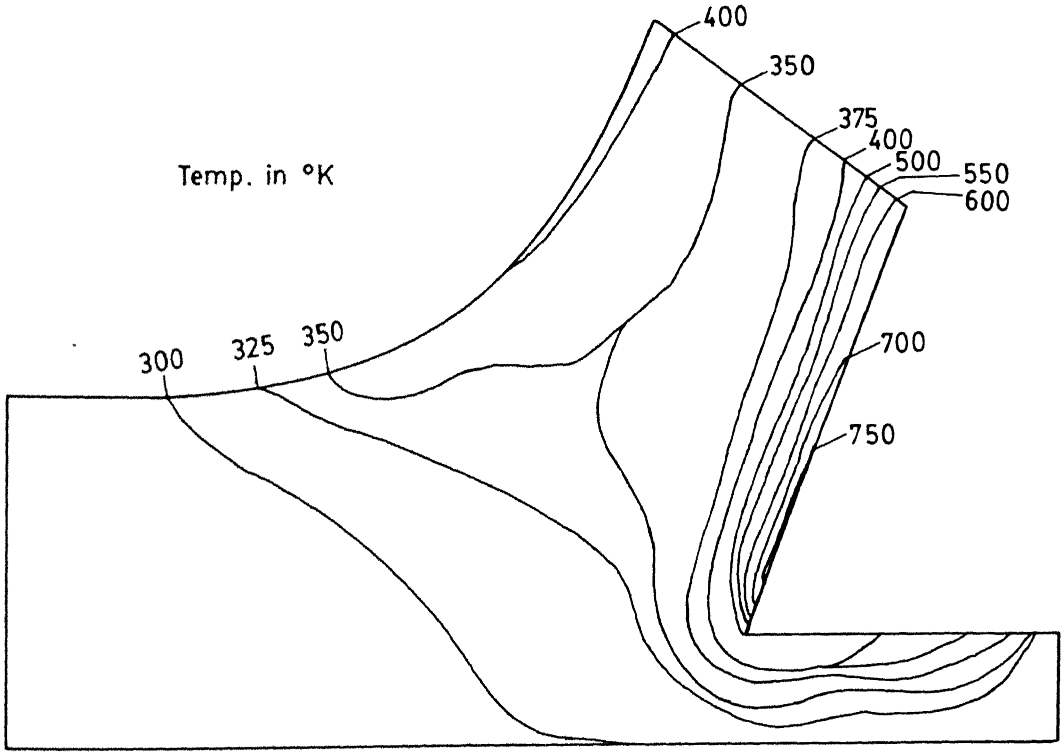


FIG. 4·5 (b) ISOTHERMAL PATTERN

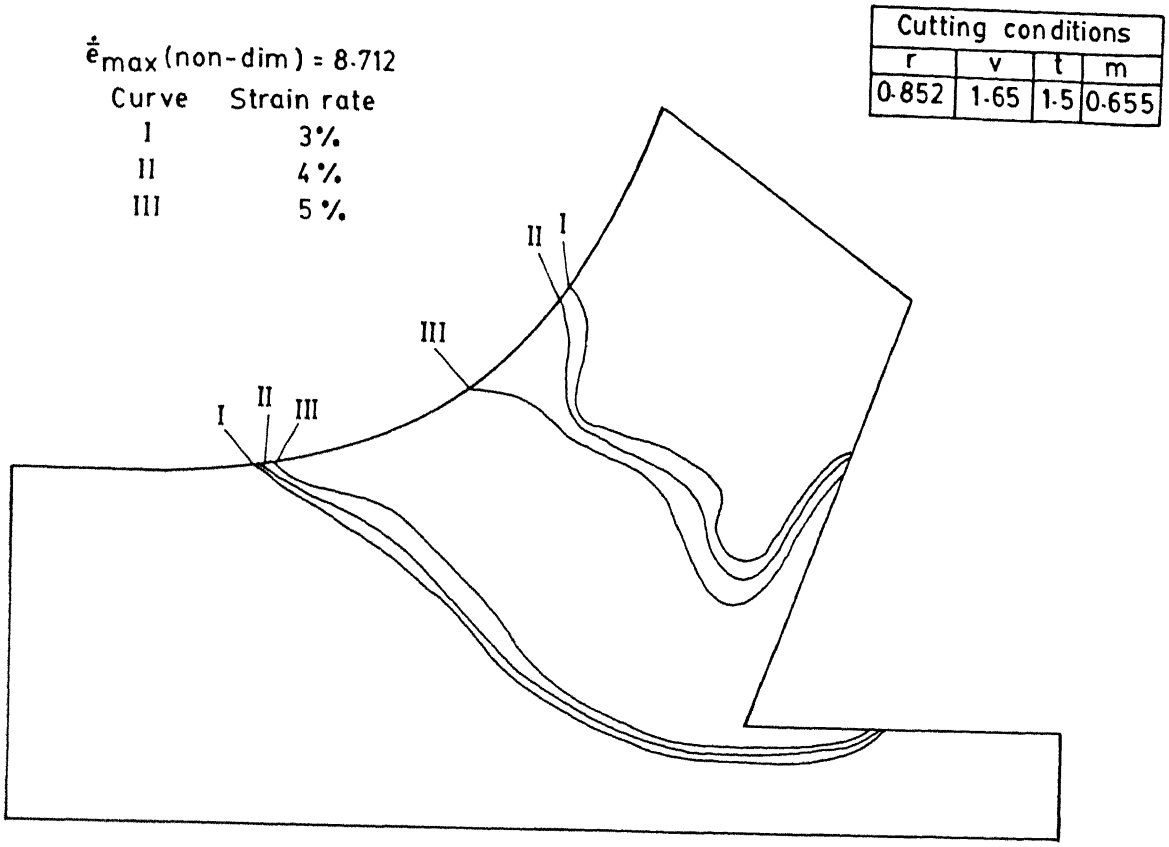


FIG. 4.6 (a) DEFORMATION PATTERN

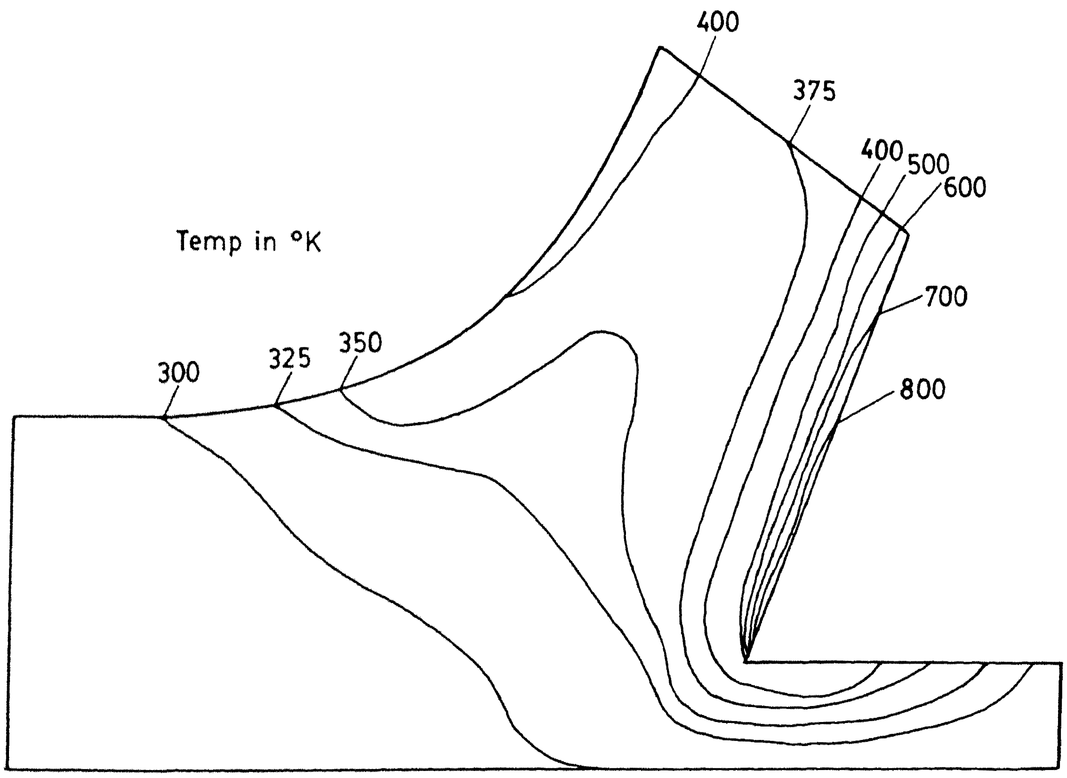


FIG. 4.6(b) ISOTHERMAL PATTERN

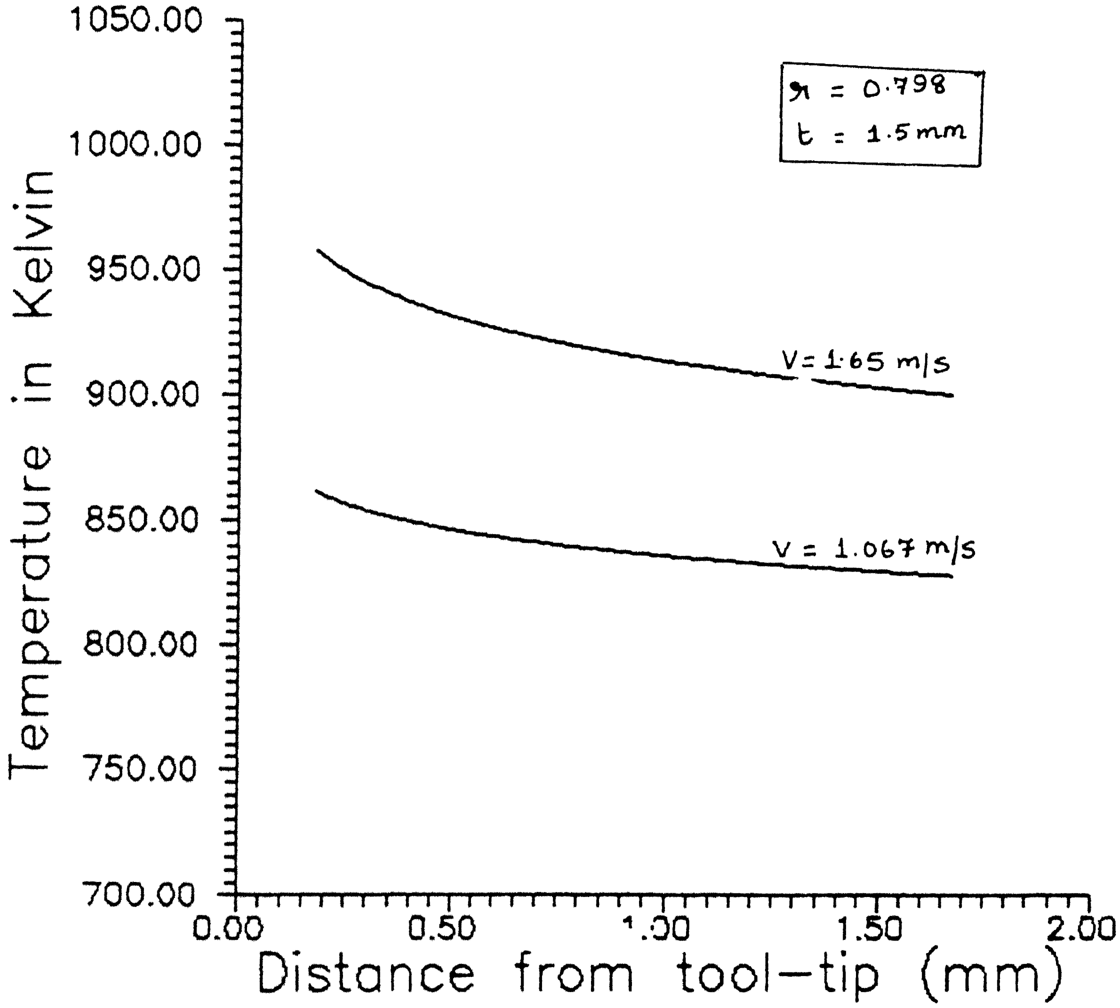


Fig. 4.7 Effect of Cutting Velocity on the Temperatures at the tool-chip interface.

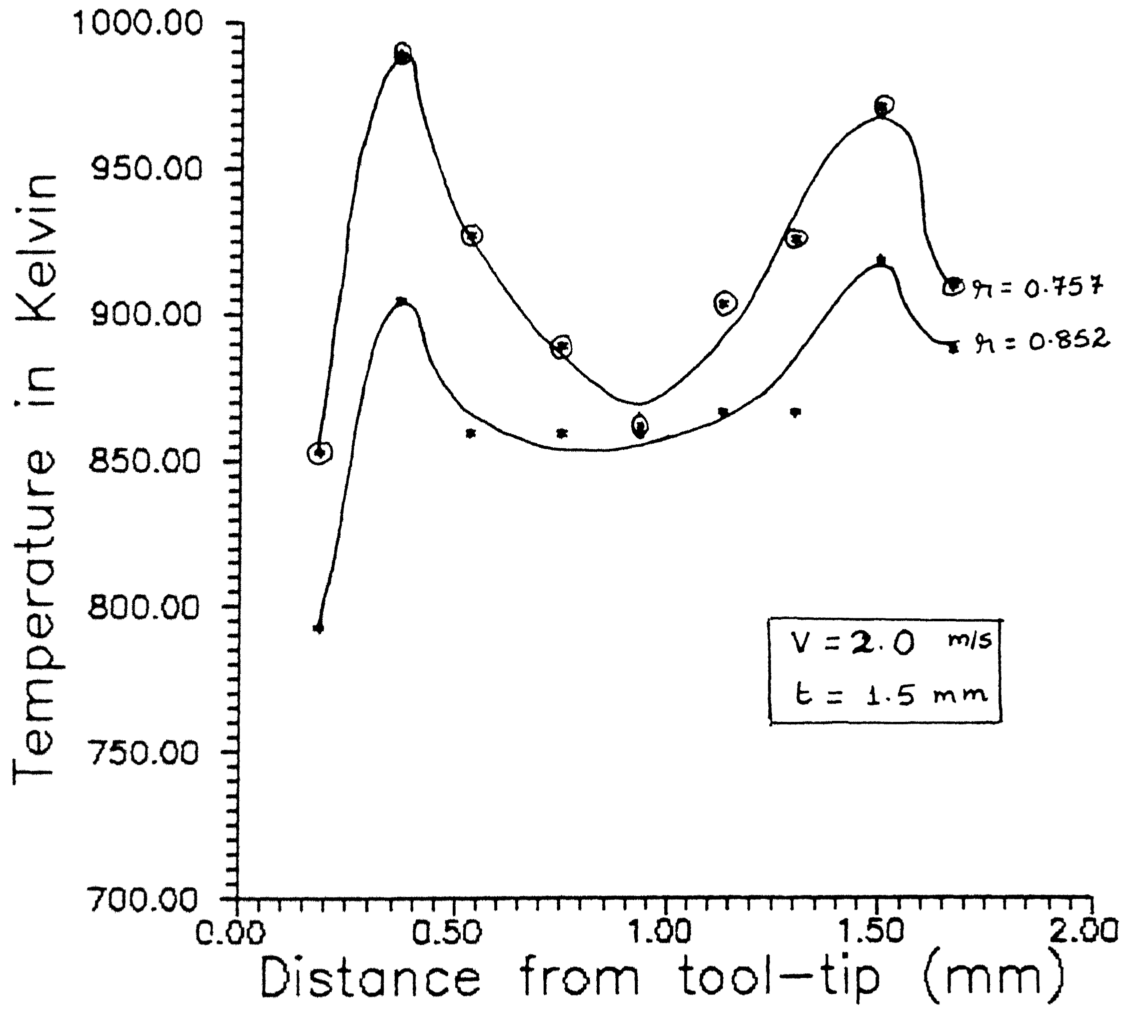


Fig. 4.8 Effect of Chip-thickness ratio on the Temperatures at the chip-tool interface.

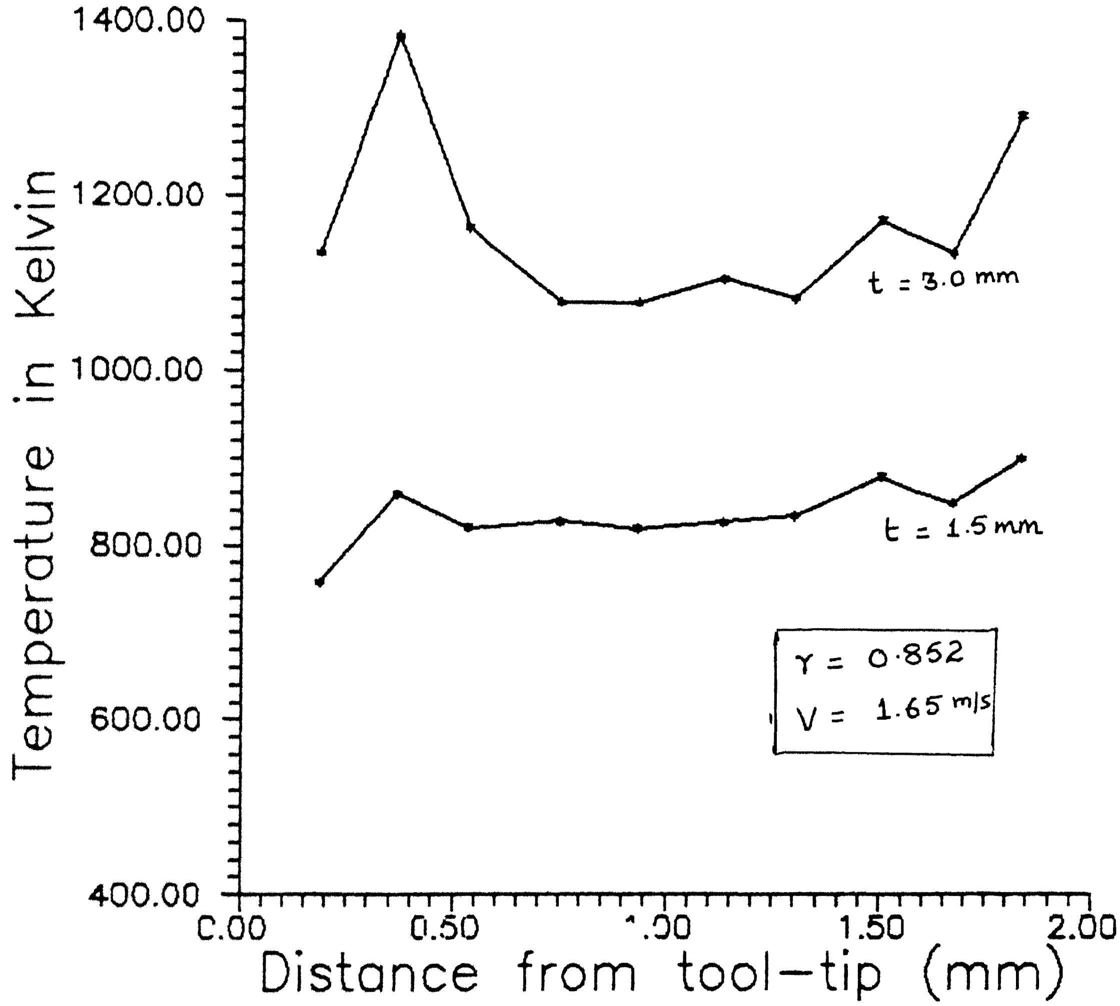


Fig. 4.9 Effect of Depth of Cut on the Temperatures at the chip-tool interface.

velocity. Consequently the temperatures are also higher, especially at the chip tool interface also seen from Fig. 4.7. It can be seen from Fig. 4.8 also that the chip thickness ratio has only marginal effects on both the temperature fields and deformation patterns. But the isothermal patterns are very much influenced by the depth of cut as also seen in Fig. 4.9. However, the primary deformation zone temperatures are affected much more than the secondary zone temperatures. This could be because the deformation may not occur uniformly across the shear plane, when the depth of cut is increased. The isotherms also seem to take irregular shapes in the case of large depth of cut, due to the same reason.

4.2 IDENTIFICATION OF THE SHEAR PLANE AND VARIATION OF PROPERTIES ALONG AND ACROSS THE PLANE

The shear plane could be identified using the available strain-rate distribution. The constant strain-rate curves for higher and higher values were plotted on either side until they approached very close to each other becoming almost parallel lines. This is shown in Fig. 4.10 where the curve IV has a strain-rate ($\dot{e}$) of 10% of the maximum. The plane AB shown in the figure can thus be said to represent the shear plane. Interestingly, the shear angle obtained in this fashion matches well with the measured one (given in table).

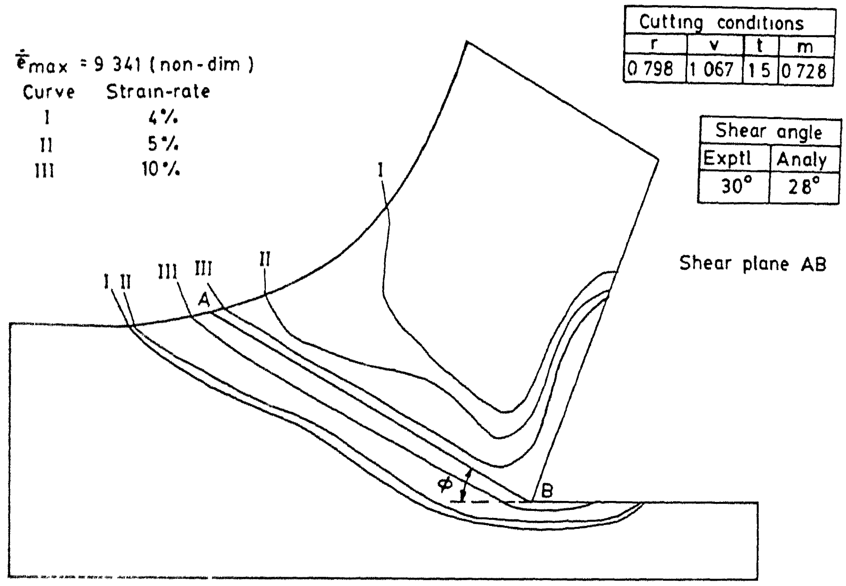


FIG 4 10 SHEAR PLANE AS OBTAINED FROM DEFORMATION PATTERN

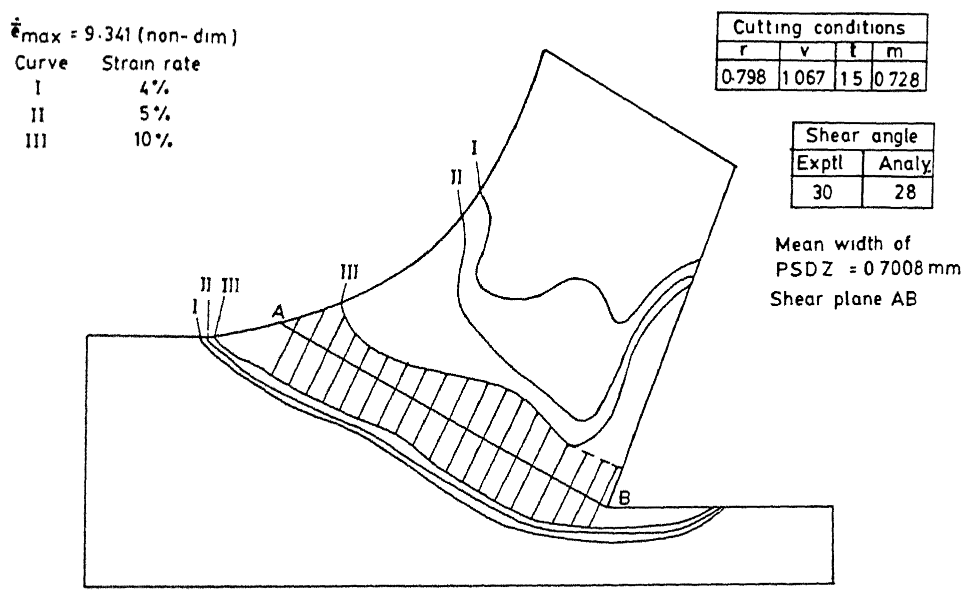


FIG 4 11 MEAN WIDTH OF PSDZ

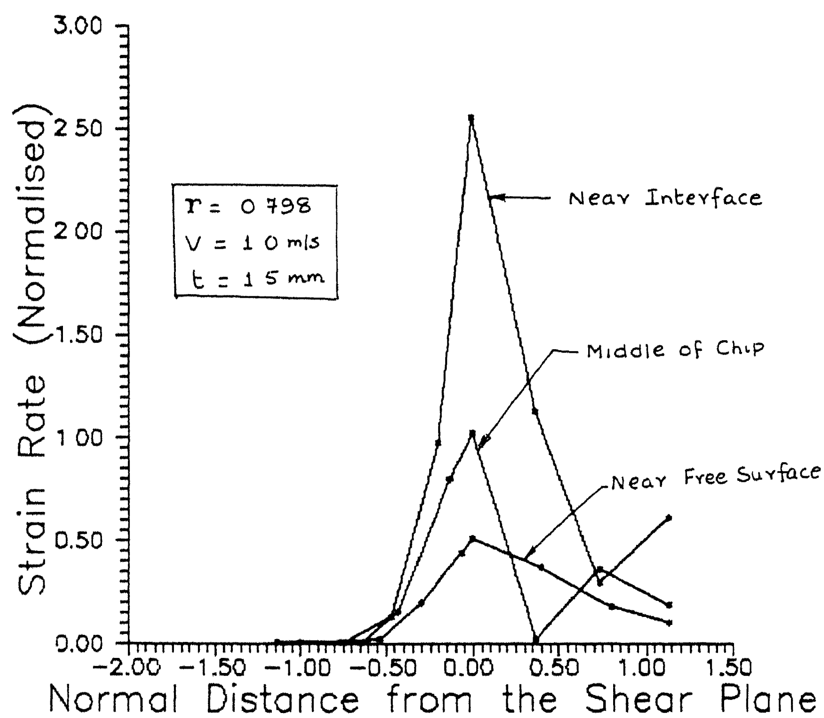


Fig. 4.12 Variation of Strain Rate Across the Shear Plane.

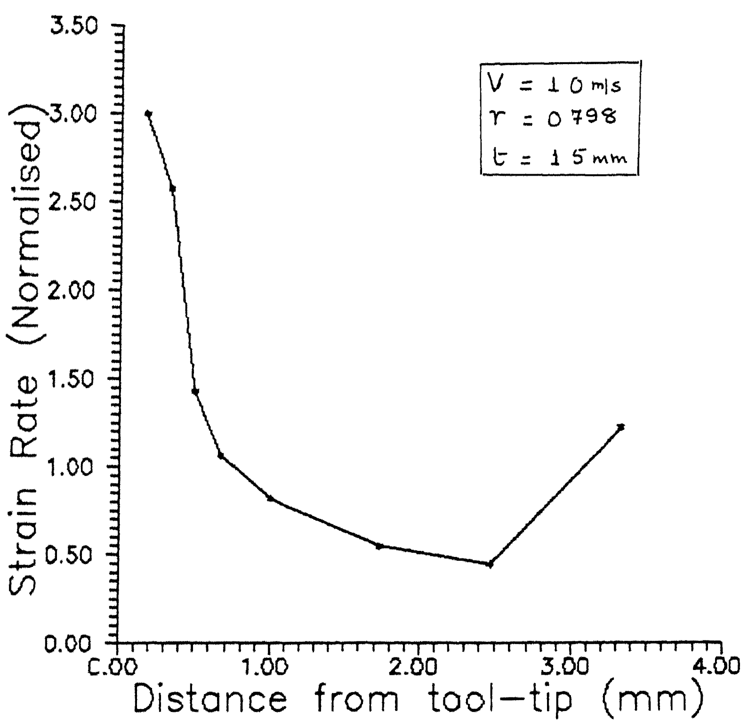


Fig. 4.13 Variation of Strain Rate Along the Shear Plane.

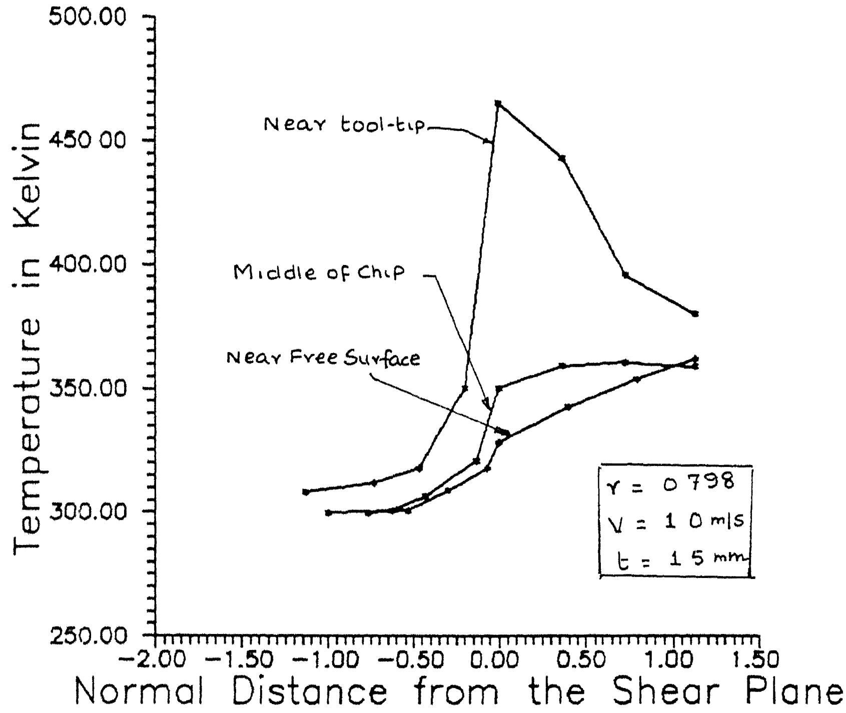


Fig. 4.14 Variation of Temperature Across the Shear Plane.

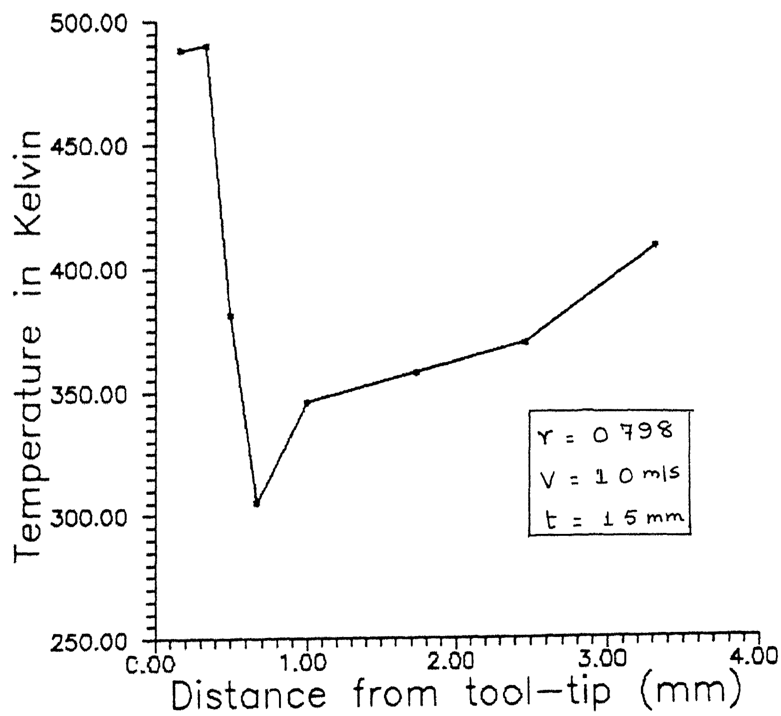


Fig. 4.15 Variation of Temperature Along the Shear Plane.

Figure 4.11 shows the same case with the shear plane and the primary deformation zone. A cut-off value of 5% has been used to indicate the boundaries of PSDZ (Curve III). To obtain the mean width of the PSDZ, normals were drawn from the shear plane to Curve III and the average length of these normals was taken as the mean width.

Often, the strain-rate and temperature variations across and along with shear plane are of particular interest. In Figs. 4.12 and 4.13, the strain-rate variation across and along the shear plane have been drawn. As can be expected, across the shear plane the strain-rate rises rapidly, reaches a maximum somewhere near the shear plane and falls rapidly again. Also the strain-rate is seen to be very high near the tool tip and small in the middle of the chip. It increases slightly as compared to the central portion, near the free surface. However, the range of variation in the strain-rate is less along the plane when compared to that across the shear plane. The same trend is observed for the temperature variation shown in Figs. 4.14 and 4.15.

4.3 VARIATION OF STRAIN-RATE NORMAL TO THE CHIP-TOOL INTERFACE

In Fig. 4.16 and 4.17, the variation of strain-rate and temperatures in a direction normal to the rake face are shown for three locations, viz. (i) near the tool-tip, (ii) middle of the contact length and (iii) end of the contact length. As can be seen the strain-rate and the temperature decrease rapidly after

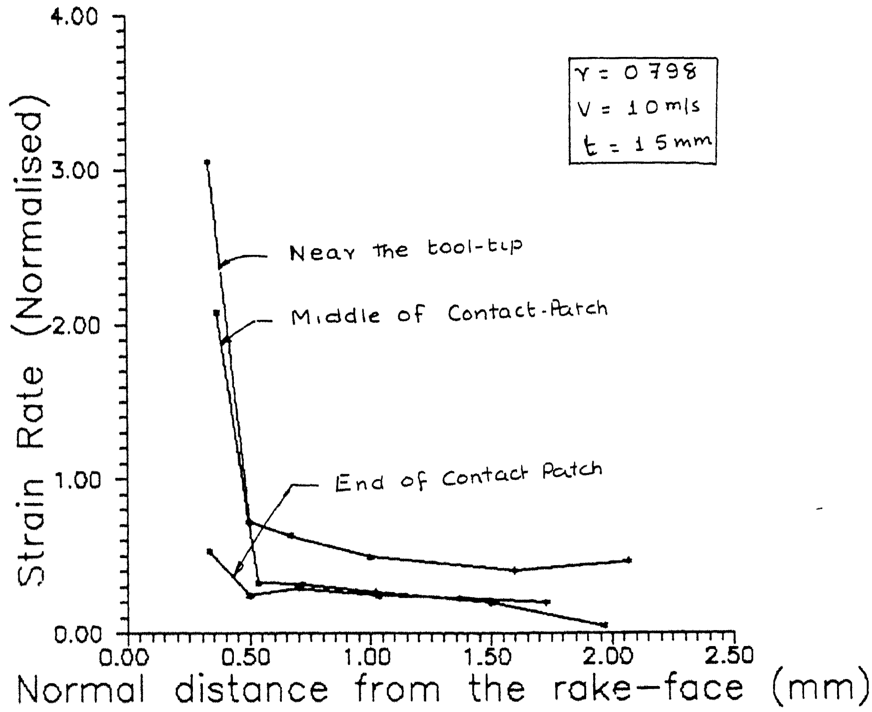


Fig. 4.16 Variation of Strain Rate Normal to the chip-tool Interface.

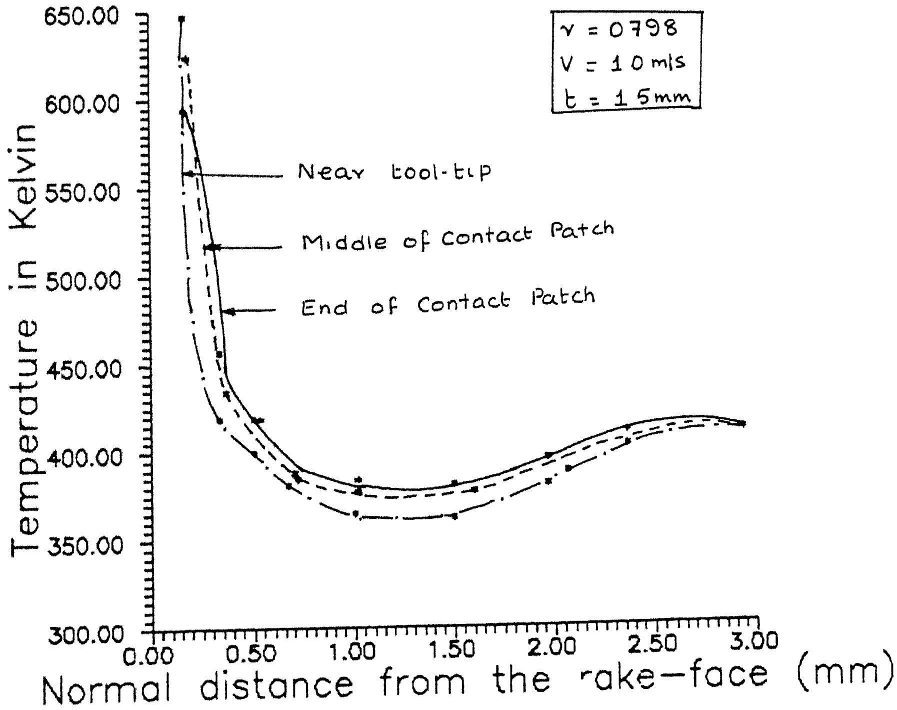


Fig. 4.17 Variation of Temperature Normal to the chip-tool Interface.

the secondary deformation zone. An idea of the structure of SSDZ can be obtained from these figures also. Similar to the determination of the width of PSDZ, the dimensions of SSDZ can also be obtained by drawing suitable cut-off curves beyond which the strain-rate and temperature gradient become very small.

The results of the present study confirmed all the well-known trends in metal cutting. However, one peculiar feature of the prediction obtained in this study is that local maxima in temperature are seen at two different locations in the chip-tool interface which has not been observed by the other researchers. One of these maximum values occurs near the tool-tip while the other is close to the location where the chip separates from the tool surface. It is believed that this feature is exhibited due to the assumption of a constant friction factor on the rake-face. With a variable friction factor (decreasing away from the tool-edge), the temperature profile is likely to exhibit a single maximum. This aspect, however, requires further detailed research.

The present study provides a theoretical methodology of estimating the dimensions of PSDZ and SSDZ, in a manner similar to that used in the analysis of photo-micrographs. The results are also helpful for locating the shear plane and determining the value of the shear angle. Further, a good estimate of the temperature variable near the tool-tip is obtained, with minimum use of empirical data.

CHAPTER 5

CONCLUSIONS AND SUGGESTIONS

From the FEM predictions of the Thermal and Deformation phenomena in Metal Cutting, the following conclusions can be drawn :

1. Viscoplastic Model for Metal Cutting can be successfully employed for an accurate prediction of the velocity and temperature fields in the vicinity of the tool-tip.

2. The deformation patterns and isotherms obtained confirm the salient features that can expected for an Orthogonal Metal Cutting situation.

3. The constant strain-rate curves plotted with cut-off values 3%, 4% and 5% give a reasonably good idea of the dimensions of the primary and secondary shear deformation zones. By increasing this cut-off value, the shear plane also can be obtained which helps in determining the mean width of PSDZ.

4. From the parametric study conducted, it is observed that the depth of cut influences the deformation and isothermal patterns much more than the cutting velocity or the chip-thickness ratio. This seems to be particularly so for the temperature variation at the tool-chip interface.

5. The deformation pattern near the chip free surface seems to have some dependence on the shape chosen to describe the free surface.

6. The choice of friction boundary condition at the chip-tool interface seems to have lot of influence on the temperatures on that boundary. A variable friction factor, decreasing away from the tool-edge is recommended.

SUGGESTIONS FOR FUTURE WORK

1. The velocity and temperature fields obtained could be further processed for predicting the required cutting forces which is one of the important input parameters.

2. A variable friction factor could be used for the friction boundary condition for improving the temperature predictions at the chip-tool interface.

3. Prediction of the contact length and the chip-thickness ratio can be attempted using the Energy Minimization principle.

4. The effects of thermal softening and work-hardening could be included.

5. The present formulation for Orthogonal cutting could be extended to Oblique cutting also.

REFERENCES

1. Merchant, M.E., "Mechanics of Metal Cutting Process", Pt. I & II, J. Appl. Phys., Vol. 16, 1945, p. 267.
2. Piispanen, V., "Theory of Formation of Metal Chips", J. Appl. Phys., Vol. 19, 1948, p. 876.
3. Palmer, W.B. and Oxley, P.L.B., "The Mechanics of Orthogonal Machining", Proc. Instn. of Mech. Engrs., Vol. 173, 1959, p. 24.
4. Okushima, K. and Hitomi, K., "On Cutting Mechanism of Soft Metals", Trans. Japan Soc., Mech. Engg., Vol. 23, No. 134, 1957, p. 674.
5. Keceioglu, D., "Shear Strain Rate in Metal Cutting & its effect on Shear Flow Stress", Trans. Am. Soc. Mech. Engrs., Vol. 80, 1958, p. 158.
6. Stevenson, M.G. and Oxley, P.L.B., "An Experimental Investigation of the Influence of the Speed and Scale on the Strain Rate in a Zone of Intense Plastic Deformation", Proc. Instn. Mech. Engrs., Vol. 184, Pt. I, No. 31, 1969-70, p. 561.
7. Black, J.T., "Flow Stress Model in Metal Cutting", ASME, J. of Engg. for Industry, Vol. 95, No. 4, 1973, p. 898.
8. Von-Turkovich, B.F., "Deformation Mechanics during Adiabatic Shear", Proc. II North American Metal Working Res. Conference, Madison, Wisconsin, 1974, p, 862.

9. Von-Turkovich, B. and Micheletti, G.F., "Flow Zone Models in Metal Cutting", 9th Int. M.T.D.R. Conf., Univ. of Birmingham, 1968.

10. Bishop, J.P.N., "An Approximate Method for determining the Temperature recorded in a Steady State Motion Problem of Plane Strain", J. Mech. Appl. Math., Vol. 9, 1956, p. 236.

11. Zienkiewicz, O.C., Jain, P.C. and Onate, E., "Flow of Solids during Forming the Extrusion, Some Aspects of Numerical Solution", Int. J. Solids Struct., Vol. 14, 1978, p. 15.

12. Zienkiewicz, O.C., Onate, E. and Heinrichs, "A General Formulation for Coupled Thermal Flow of Metals Using Finite Element", Int. J. Num. Meth. in Engg., Vol. 17, 1981, p.1497.

13. Zienkiewicz, O.C. and Godbole, P.N., "Flow of Plastic and Visco-plastic Solids with Special reference to Extrusion and Forming Processes", Int. J. for Num. Methods in Engg., Vol.8.

14. Tay, A.O., Stevenson, M.G. and de Vahl Davis, G., "Using the Finite Element Method to Determine Temperature Distributions in Orthogonal Machining", Proc. Inst. of Mech. Engrs., Vol.188, 1974, pp. 627-638.

15. Tay, A.O., Stevenson, M.G., deVahl Davis, G., and Oxley, P.L.B., "A Numerical Method for Calculating Temperature Distributions in Machining, from Force and Shear Angle Measurements," Int. J. of Machine Tool Design and Research, Vol. 16, 1976, pp. 335-349.

16. **Stevenson, M.G., Wright, P.K. and Chow, J.G.,** "Further Developments in Applying the Finite Element Method to the Calculation of Temperature Distributions in Machining and Comparisons with Experiment", **Trans. ASME, J. of Engg. for Industry,** Aug. '83, Vol. 105, pp. 149-154.

17. **Muraka, P.D., Barrow, G., and Hinduja, S.,** "Influence of the Process Variables on the Temperature Distribution in Orthogonal Machining Using the Finite Element Method", **Int. J. of Mech. Sci.,** Vol. 21, 1979, pp. 445-456.

18. **H.S. Balaji,** "An Application of FEM to the Estimation of Temperature Field in Machining with Coated Carbide Tools", **M.Tech. Thesis,** Dept. of Mech. Engg., IIT, Kanpur.

19. **Oslon, M.D.,** "Variational Finite Element Methods for Two-dimensional and Axisymmetric Navier-Stokes Equations", **Finite Element in Fluids,** Vol. 1, **Viscous Flow and Hydrodynamics,** (Ed) Gallagher R.H. et al., Jogn Wiley, London, 1975.

20. **Taylor, C. and Hughes, T.G.,** "Finite Element Programming of the Navier-Stokes Equations", Ist Ed., 1981, Pineridge Press Ltd. U.K.

APPENDIX A

MATERIAL PROPERTIES

Density (ρ) $= 7880\ Kg/m^3$

Specific heat of workpiece material (C_p) $= 500\ J/Kg.K$

Thermal conductivity of workpiece material (k) $= 50\ W/mK$

Thermal conductivity of tool material (k_t) $= 25\ W/mK$

Uniaxial Yield Stress (σ_y) $= 300\ MPa$

HEAT TRANSFER CONDITION

Overall heat transfer coefficient (h) $= 50\sqrt{v}$

Ambient temperature (T_f, T_{amb}) $= 300\ K$

CUTTING PARAMETER

S.No.	Cutting Velocity (m/s)	Chip thickness Ratio	Friction factor	Contact Length
1.	1.0	0.757	0.637	1.85
2.	1.65	0.798	0.728	1.85
3.	1.20	0.852	0.655	1.47

APPENDIX B

DERIVATION OF EQUATION FOR VISCOSITY

For a viscous, incompressible fluid we can write the constitutive relation in the form

$$\sigma_{ij} = \sigma\delta_{ij} + 2\mu\dot{e}_{ij}; \quad \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{1}$$

in which σ is the mean stress and $\dot{e}_{ij}$ are the strain rates. Alternatively we can rewrite equation (1) as

$$\dot{e}_{ij} = \frac{1}{2\mu} s_{ij} \tag{2}$$

in which s_{ij} are the deviatoric components of stress.

For a visco-plastic, 'associated' material we can write with some degree of generality

$$e_{ij} = \frac{1}{\bar{\mu}} \langle F \rangle \frac{\partial F}{\partial \sigma_{ij}} \tag{3}$$

where $\bar{\mu}$ is a constant 'pseudo-viscosity', and

$$F = F(\sigma_{ij}) = 0 \tag{4}$$

represents the plastic yield condition. Further we use the following notation which ensures no plastic flow when stresses are below yield

$$\langle F\rangle = F \quad \text{of} \quad F > 0 \quad \text{and} \quad \langle F\rangle = 0 \quad \text{of} \quad F \leq 0 \tag{4}$$

Clearly as the constant $\bar{\mu} \to 0$, the equation (3) specifies the behaviour of an ideally plastic mateiral. Specializing the above relationship to the von Mises criterion for which

$$F = \sqrt{\left(\frac{1}{2}\, s_{ij} s_{ij}\right)} - \frac{1}{3}\, Y \tag{5}$$

where Y is a constant representing some yield parameter (here as the uniaxial test yield value) we can write

$$\frac{\partial F}{\partial \sigma_{ij}} = \frac{1}{2}\, \frac{1}{\sqrt{\left(\frac{1}{2}\, s_{ij} s_{ij}\right)}} \tag{6}$$

If flow occurs, by definition (3) $F \geq 0$ and we can identify expressions (2) and (3). Using also equations (5) and (6) we can write

$$\frac{1}{\mu} = \frac{1}{\bar{\mu}}\, \frac{\sqrt{\left(\frac{1}{2}\, s_{ij} s_{ij}\right) - \left(\frac{1}{\sqrt{3}}\right) Y}}{\sqrt{\left(\frac{1}{2}\, s_{ij} s_{ij}\right)}} \tag{7}$$

From equation (2), we can also write the identity

$$\sqrt{\left(\frac{1}{2}\, s_{ij} s_{ij}\right)} = \mu \sqrt{\left(2 \dot{e}_{ij} \dot{e}_{ij}\right)} = \mu \dot{\bar{e}} \qquad (8)$$

in which $\dot{\bar{e}}$ is the second strain invariant and inserting this into equation &) we obtain finally an equivalent viscosity as

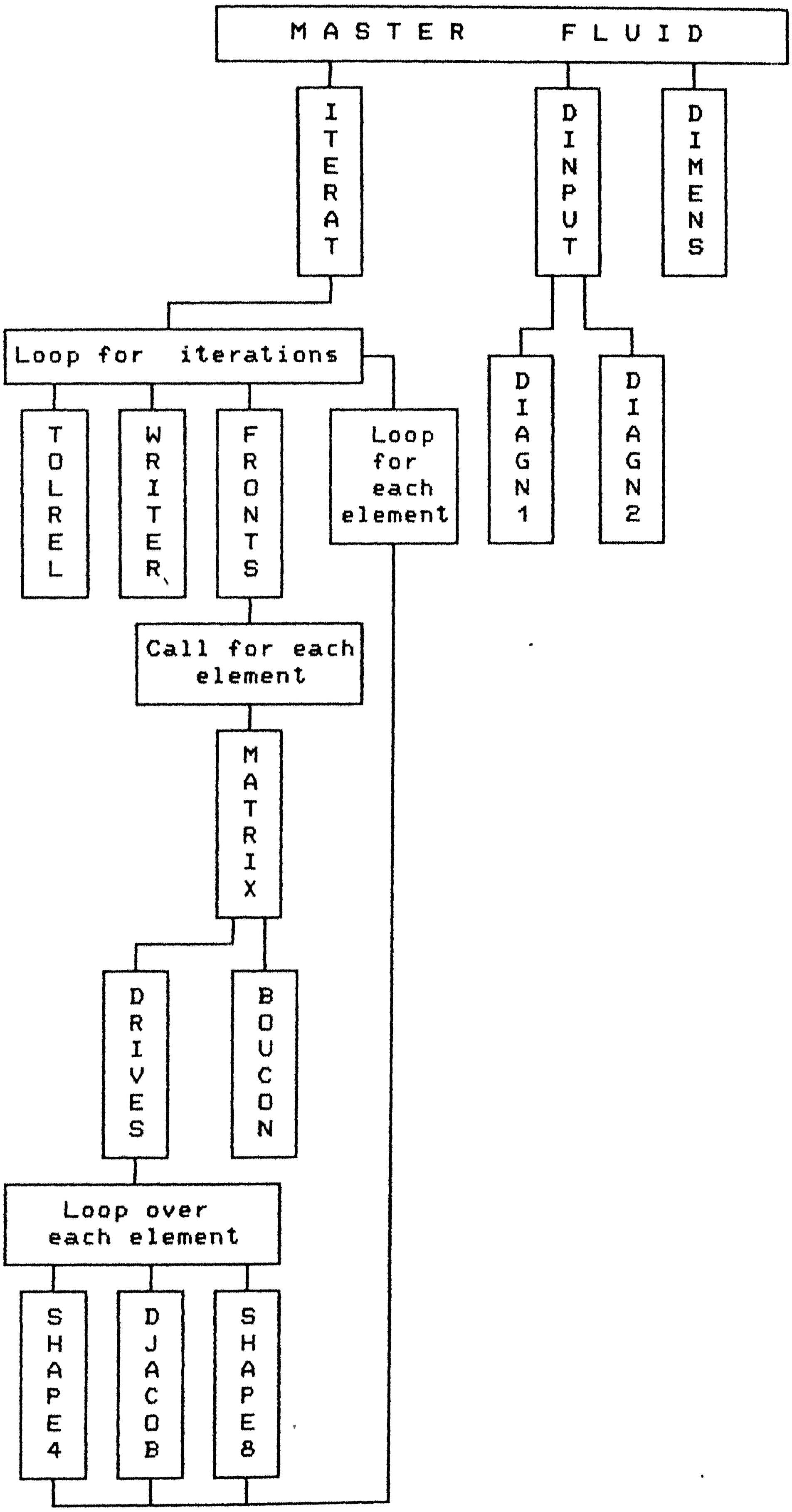
$$\mu = \frac{\left(\frac{1}{\sqrt{3}}\right) Y + \bar{\mu} \dot{\bar{e}}}{\dot{\bar{e}}} \qquad (9)$$

in which the parameter $\dot{\bar{e}}$ describes an 'effective' strain rate. Relations (9) are a particular case of a non-Newtonian fluid where in general we can write

$$\mu = \mu(\dot{\bar{e}}) \qquad (10)$$

and where the functional form is specified in an experimental manner. As can easily be observed the visco-plastic model is a generalization of the special case ofa Bingham fluid.

APPENDIX C


MASTER FLUID
ITERAT
DINPUT
DIMENS
Loop for iterations
DIAGN1
DIAGN2
TOLREL
WRITER
FRONTS
Loop for each element
Call for each element
MATRIX
DRIVES
BOUCON
Loop over each element
SHAPE4
DJACOB
SHAPE8

A P P E N D I X D

```
C*************************************************************
C       MASTER FLUID
C*************************************************************
C__
C_      EXTERNAL SUBROUTINES
C_      DIMENS
C       DINPUT
C_      DRIVES
C__     ITERAT
C__
C*************************************************************
        COMMON/ARC1/POSGP(3),WEIGP(3)
        COMMON/ARC2/LNODS(105,8),COORD(366,2),NADFM(366),NODFM(366)
        COMMON/ARC3/VARB1(1229),VARB2(1229)
        COMMON/ARC35/EPSN(105,9)
        DIMENSION EQRHS(1229)
        DIMENSION LBOUD(1229),LHEDV(150),PNORM(150),BOUDV(1229)
        DIMENSION GFLUM(150,150)
        OPEN(UNIT=20,FILE='s3v3.inp')
        OPEN(UNIT=40,FILE='var.inp')
        OPEN(UNIT=24,FILE='s3v3.out')
        OPEN(UNIT=35,FILE='s3v3v.out')
        OPEN(UNIT=28,FILE='s3v3e.out')
C
C***    SET DYNAMIC DIMENSION VALUES
C
        CALL DIMENS(MELEM,MFRON,MPOIN,MTOTV)
C
C
C***    READ IN ALL PROBLEM DATA.
C
        CALL DINPUT(IAXSY,BOUDV,LBOUD,MELEM,
     !  MPOIN,MTOTV,NBCON,NDOFM,NELEM,NEVAB,NGAUS,NITER,
     !  NNODL,NNODP,NPOIN,NTOTV,RELAX,TOLER,
     !  XFORC,YFORC)
C
C***    CALCULATE THE SHAPE FUNCTION AND DERIVATIVE VALUES
C
C
C***    SET UP EQUATIONS AND ITERATE UNTILL SOLUTIONS CONVERGE.
C
        CALL ITERAT(IAXSY,GFLUM,BOUDV,LBOUD,LHEDV,PNORM,EQRHS,
     !  MELEM,MFRON,MPOIN,MTOTV,NBCON,NDOFM,
     !  NELEM,NEVAB,NGAUS,NITER,NNODL,NNODP,NPOIN,NTOTV,
     !  RELAX,TOLER,XFORC,YFORC)
        WRITE(28,100)
100     FORMAT(20X'STRAIN RATE DISTRIBUTION',//,
     !  'KELEM',45X,'GAUSS POINTS',/,
     !  18X,'1',12x,'2',12X,'3',12x,'4',10X,'5',10X,'6',10X,'7',
     !  12X,'8',12X,'9',/)
        DO 200 I = 1,MELEM
        WRITE(28,111) I,(EPSN(I,J),J=1,9)
111     FORMAT(I3,5x,9(E10.4,2X))
200     CONTINUE
        type*,'MAIN OVER'
        STOP
        END
C*************************************************************
        SUBROUTINE DIMENS(MELEM,MFRON,MPOIN,MTOTV)
C
C***    SET DYNAMIC DIMENSION ARREY SIZES.
C
        MELEM=105
        MFRON=150
        MPOIN=366
```

```
      RETURN
      END
C******************************************************************
      SUBROUTINE DINPUT(IAXSY,BOUDV,LBOUD,
     ! MELEM,MPOIN,MTOTV,NBCON,NDOFM,NELEM,NEVAB
     ! ,NGAUS,NITER,NNODL,NNODP,NPOIN,NTOTV,RELAX,
     ! TOLER,XFORC,YFORC)
      COMMON/ARC2/LNODS(105,8),COORD(366,2),NADFM(366),NODFM(366)
      DIMENSION LBOUD(MTOTV),BOUDV(MTOTV)
      COMMON/ARC3/VARB1(1229),VARB2(1229)
      COMMON/ARC10/CUTVEL,DCUT,BSTAR
      COMMON/ARC12/HSTAR,TFSTAR,FRIFAC,PI,EN,COSA,SINA,CONRAT
      COMMON/ARC13/USTAR,PECLET,SIGMA,RHO,MBOUEL
      COMMON/ARC14/ IBEL(50),ISURF(50),ISIDE(50)
      COMMON/ARC30/ANGLN(366)
C      OPEN(UNIT=20,DEVICE='DSK',FILE='SMALL.INP')
C      OPEN(UNIT=24,DEVICE='DSK',FILE='SMALL.OUT')
C
C***  GIVE VALUES TO THOSE VARIABLES WHICH CAN NOT BE CHANGED
      RHO=7820.0
      NDOFM=4
      NEVAB=28
      NGAUS=3
      NNODL=4
      NNODP=8
C
C***  READ IN EACH LINE & ECHO IMMEDIATELY
C
C     READ(20,1000)TITLE
C1000 FORMAT(12A6)
C     WRITE(24,2000) TITLE
C2000 FORMAT(1H1,//1X,12A6)
      READ(20,*)IAXSY,NELEM,NITER,NPOIN,NRPON
      pause 1
      WRITE(24,*)IAXSY,NELEM,NITER,NPOIN,NRPON
      WRITE(24,*)NELEM,NPOIN,NNODP
C2010  FORMAT(//13H CONTROL DATA,/13H ***********,//
C    ! 8H IAXSY =,I4,4X,8H NELEM =,I4,4X,8H NITER =,I4,4X,
C    ! 8H NPOIN =,I4,4X,8H NRPON =,I4)
      READ(20,*)NICON,NBCON,NEBCN,NNBCN
      pause 2
      WRITE(24,2020)NICON,NBCON,NEBCN,NNBCN
2020  FORMAT(8H NICON=,I4,4X,8H NBCON=,I4,4X,
     ! 8H NEBCN=,I4,4X,8H NNBCN=,I4)
C
C***  CHECK INITIAL DATA
C
     CALL DIAGN1(NBCON,NEBCN,NELEM,NICON,NNBCN,NPOIN,NRPON)
C
C***  READ FLOW PARAMETER
C
      READ(20,*)RELAX,TOLER,XFORC,YFORC
      pause 3

      WRITE (24,2030)RELAX,TOLER,XFORC,YFORC
2030  FORMAT(//20H PHYSICAL PROPERTIES,/20H ******************,
     ! //4X,8H RELAX =,F10.5,4X,8H TOLER =,F10.5,
     ! /8H XFORC =,F10.5,4X,8H YFORC =,F10.5)
      READ(20,*)EN,GAMMA,SIGMAY
      pause ' 4'
      WRITE(24,21153)EN,GAMMA,SIGMAY
21153 FORMAT(//22H VISCOSITY PARAMETERS,/22H ********************,
     ! //4H EN=,F10.5,4X,7H GAMMA=,E10.5,4X
     ! ,8H SIGMAY=,E10.5)
      READ(20,*) CUTVEL,DCUT,RAKE,FRIFAC
```

```
      WRITE(5,*) MSURF
      READ(20,*)MELS1,MELS2,MELS3,MELS4,MELS5,MELS6,MELS7,MELS8
      pause '12'
      WRITE(24,*)MELS1,MELS2,MELS3,MELS4,MELS5,MELS6,MELS7,MELS8
      WRITE(5,*)MELS1,MELS2,MELS3,MELS4,MELS5,MELS6,MELS7,MELS8
      MBOUEL=MELS1+MELS2+MELS3+MELS4+MELS5+MELS6+MELS7+MELS8
      DO 9 II=1,MBOUEL
      JJ = II
      READ(20,*) JJ, ISURF(II),IBEL(II),ISIDE(II)
      WRITE(24,*) JJ, ISURF(II),IBEL(II),ISIDE(II)
      WRITE(5,*) JJ, ISURF(II),IBEL(II),ISIDE(II)
9     CONTINUE
      pause '13 mbouel'
      WRITE(24,*) MBOUEL
      WRITE(5,*) MBOUEL
C***  SET UP THE VECTOR GIVING THE D.O.F. AT EACH NODE
C
      ITEMP=NNODP-1
      DO 40 IELEM=1,NELEM
      DO 40 INODP=1,ITEMP,2
      NODFM(LNODS(IELEM,INODP))=NDOFM
      NODFM(LNODS(IELEM,INODP+1))=NDOFM-1
40    CONTINUE
C
C***  INTERPOLE MIDSIDE NODE COORD WHERE NOT KNOWN
C     SKIPPING THE INTERPOLATION DUE TO AVAILABILITY OF COORDINATES
      GOTO 71
      DO 70 IELEM=1,NELEM
      DO 60 INODP=2,NNODP,2
      IPOIN=LNODS(IELEM,INODP)
      TEMPY=ABS(COORD(IPOIN,1))+ABS(COORD(IPOIN,2))
      IF(TEMPY.NE.0.0) GO TO 60
      JPOIN=LNODS(IELEM,INODP-1)
      KNODP=INODP+1
      IF(KNODP.GT.NNODP)KNODP=1
      KPOIN=LNODS(IELEM,KNODP)
      DO 50 IDIME=1,2
      COORD(IPOIN,IDIME)=(COORD(JPOIN,IDIME)+COORD(KPOIN,IDIME))*.5
50    CONTINUE
60    CONTINUE
70    CONTINUE
71    CONTINUE
C
C***  SET UP VECTOR WITH THE FIRST D.O.F.AT EACH NODE
C
      NADFM(1)=1
      DO 80 IPOIN=2,NPOIN
      NADFM(IPOIN)=NADFM(IPOIN-1)+NODFM(IPOIN-1)
80    CONTINUE
      NTOTV=NADFM(NPOIN)+NODFM(NPOIN)-1
      TYPE*,NTOTV
      pause 'ntotv'
C***  INITIALISE REMAING ARRAYS.
C
      DO 90 ITOTV=1,NTOTV
      LBOUD(ITOTV)=0
      BOUDV(ITOTV)=0.0
      VARB1(ITOTV)=0.0
      VARB2(ITOTV)=0.0
90    CONTINUE
      DO 868 ITOTV = 1,MTOTV
      READ(40,*)VARB2(ITOTV)
      TYPE*,VARB2(ITOTV),VARB1(ITOTV)
868   CONTINUE
C
```

```
C
      IF(NICON.EQ.0)GO TO 110
      DO 100 IICON=1,NICON
      READ(20,*)IPOIN,IDOFM,TEMPY
      IF(IDOFM.GT.1)IDOFM=NODFM(IPOIN)
      JTOTV=NADFM(IPOIN)+IDOFM-1
      VARB1(JTOTV)=TEMPY
      VARB2(JTOTV)=TEMPY
100   CONTINUE
110   CONTINUE
      pause '110'
C
C***  CHECK NODAL CONNECTION & COORDINATES
C
      CALL DIAGN2(MELEM,MPOIN,NELEM,NICON,
     !  NNODP,NPOIN,NTOTV)
C
C***  READ IN BOUNDRY CONDITION
      pause'bound.con'
      WRITE(24,2090)
2090  FORMAT(//20H BOUNDARY CONDITIONS//20H ******************,//
     !  35H NODE   U-FIXED  PRESSURE   V-FIXED)
      pause'reading bc.'
      WRITE(5,*)NBCON
      pause'nbcon'
      DO 175 IBCON=1,NBCON
      READ(20,*)IPOIN,IDOFM,BVALU
      IF(IDOFM.EQ.4) BVALU = BVALU/TREF
      write (5,*) IBCON,IPOIN
      ITOTV=NADFM(IPOIN)+IDOFM-1
      IF(NODFM(IPOIN).EQ.4) GOTO 238
      IF((IDOFM+1).GT.NODFM(IPOIN))ITOTV=ITOTV-1
238   LBOUD(ITOTV)=1
      BOUDV(ITOTV)=BVALU
      GO TO (140,150,160,165)IDOFM
140   CONTINUE
      WRITE(24,2100)IPOIN,BVALU
2100  FORMAT(I5,F9.4)
      GO TO 170
150   CONTINUE
      WRITE(24,2110)IPOIN,BVALU
2110  FORMAT(I5,10X,F9.4)
      IF(NODFM(IPOIN).NE.(IDOFM+1)) GO TO 170
      WRITE(24,2120)IBCON,IPOIN
2120  FORMAT(/27H ERROR!! BOUNDRY CONDITION,I4,
     !  42H DEFINES A PRESSURE AT MIDSIDE NODE NUMBER,I4)
      stop
160   CONTINUE
      WRITE(24,2130)IPOIN,BVALU
2130  FORMAT(I5,20X,F9.4)
      GOTO 170
165   CONTINUE
      WRITE(24,2140) IPOIN,BVALU
2140  FORMAT(I5,30X,F9.4)
      JTOTV = ITOTV
170   WRITE(5,*) JTOTV,LBOUD(ITOTV),BOUDV(ITOTV),IPOIN
175   CONTINUE
77799 type*,'DINPUT OVER'
      DO 990 ITOTV = 1,NTOTV
      JTOTV = ITOTV
      WRITE(24,3000) JTOTV,LBOUD(ITOTV),BOUDV(ITOTV)
3000  FORMAT(1x,'ITOTV=',I3,2x,'LBOUD(ITOTV)=',I3,2x,
     !  'BOUDV(ITOTV)=',F10.4)
990   CONTINUE
      DO 300 I=1,MPOIN
```

```
300     CONTINUE
        I1 = 150
        J1 = 264
        L1 = 144
        DO 350 I=1,6
        I1 = I1 + 1
        J1 = J1 + 1
        L1 = L1 + 1
        ANGLN(I1) = 20.0
        ANGLN(J1) = 20.0
        ANGLN(L1) = 90.0
350     CONTINUE
        ANGLN(82) = 90.0
        ANGLN(83) = 90.0
        ANGLN(84) = 90.0
        ANGLN(144) = 90.0
        ANGLN(253) = 85.0
        ANGLN(254) = 84.0
        ANGLN(255) = 77.0
        ANGLN(256) = 75.0
        ANGLN(257) = 68.5
        ANGLN(258) = 67.5
        ANGLN(259) = 66.25
        ANGLN(260) = 65.0
        ANGLN(261) = 63.0
        ANGLN(262) = 60.5
        ANGLN(263) = 54.0
        ANGLN(264) = 49.0
        ANGLN(271) = 20.0
        ANGLN(272) = 20.0
        ANGLN(337) = 45.5
        ANGLN(338) = 42.0
        ANGLN(339) = 41.0
        ANGLN(340) = 37.0
        ANGLN(341) = 30.5
        ANGLN(342) = 20.0
        ANGLN(343) = 20.0
        ANGLN(344) = 20.0
        ANGLN(363) = 90.0
        ANGLN(364) = 90.0
        ANGLN(365) = 90.0
        ANGLN(366) = 90.0
        DO 400 KKL = 1,MPOIN
        LL = KKL
        type*,LL,ANGLN(KKL)
400     CONTINUE
        RETURN
        END
C*******************************************************************
        SUBROUTINE DIAGN1(NBCON,NEBCN,NELEM,NICON,NNBCN,NPOIN,NRPON)
        DIMENSION NECHO(150),NEROR(8)
C
C***    SCRUTINY OF CONTROL DATA.
        DO 10 IEROR=1,8
10      NEROR(IEROR)=0
C
C***    SCRUTINISE CONTROL DATA AND PRINT ERROR MESSAGE.
C
        IF(NPOIN.LE.0)                  NEROR(1)=1
        IF(NRPON.LE.0.OR.NRPON.GT.NPOIN)NEROR(2)=1
        IF(NELEM.LE.0)                  NEROR(3)=1
        IF(NPOIN.GT.NELEM*8)            NEROR(4)=1
        IF(NBCON.GT.NPOIN*4)            NEROR(5)=1
        IF(NICON.GT.NPOIN*4)            NEROR(6)=1
        IF(NEBCN.GT.NELEM)              NEROR(7)=1
```

```
C
C***    CHECK ON ERROR-IF UNITY PRINT.
C
        JEROR=0
        DO 20 IEROR=1,8
        IF(NEROR(IEROR).EQ.0)GO TO 20
        JEROR=1
C
C***    WRITE OUT ERROR NUMBER.
C
        WRITE(24, *)IEROR
2000    FORMAT(/,10X,33H CONTROL PARAMETER ERROR****ERROR,I5)
20      CONTINUE
        IF(JEROR.EQ.0)RETURN
C
C***    LIST DATA REMAINING AFTER CONTROL PARAMETERS.
C
        WRITE(24, *)
2010    FORMAT(/10X,37HDATA FOLLOWING ERROR IN CONTROL CARDS/)
30      CONTINUE
        READ(20,1000)NECHO
1000    FORMAT(150A1)
        WRITE(24,*)NECHO
2020    FORMAT(10X,80A1)
C        GO TO 30
        type*,'DIAG1 OVER'
        RETURN
        END
C***************************************************************
        SUBROUTINE DIAGN2(MELEM,MPOIN,NELEM,NICON
     !  ,NNODP,NPOIN,NTOTV)
        DIMENSION NECHO(150),NEROR(13)
        COMMON/ARC2/LNODS(105,8),COORD(366,2),NADFM(366),NODFM(366)
C
C***    SCRUTINISE NODAL POINT AND ELEMENT DATA.
C
C***    INITIALISE ERROR ARRAY.
C
        DO 10 IEROR=1,13
        NEROR(IEROR)=0
10      CONTINUE
C
C***    CHECK PHYSICAL PROPERTIES.
C
C       IF(DENS.LE.0.0.OR.VISCY.LE.0.0)NEROR(9)=1
C
C***    CHECK NODAL COORDINATES.
C***    CHECK FOR TWO IDENTICAL COORDINATES.
C
        DO 40 IPOIN=2,NPOIN
        JPOIN=IPOIN-1
        DO 30 KPOIN=1,JPOIN
        IF(COORD(IPOIN,1).NE.COORD(KPOIN,1).OR.COORD(IPOIN,2).NE.
     !  COORD(KPOIN,2))GO TO 20
        NEROR(10)=1
20      CONTINUE
30      CONTINUE
40      CONTINUE
C
C***    CHECK ELEMENT NODAL NUMBERS
C***    A--REPITITION OF ANODE NUMBERIN ONE ELEMENT
C***    B--NOD NUMBER OUTSIDE PERMISIBLE BOUNDS.
C
        DO 50 IPOIN=1,NPOIN
        DO 50 IELEM=1,NELEM
```

```
      DO 50 INODE=1,NNODP
      IF(LNODS(IELEM,INODE).NE.IPOIN)GO TO 50
      LCONT=LCONT+1
      IF(LCONT.GT.1)NEROR(11)=1
50    CONTINUE
C      GOTO 60
      DO 60 IELEM=1,NELEM
      DO 60 INODE=1,NNODP
      IF(LNODS(IELEM,INODE).LE.0.OR.LNODS(IELEM,INODE).GT.NPOIN)
     ! NEROR(12)=1
60    CONTINUE
C
C***  CHECK ON NUMBER OF INITIAL CONDITIONS
C
      IF(NICON.GT.NTOTV)NEROR(13)=1
C
C***  WRITE OUT ERROR NUMBER
C
      JEROR=0
      DO 70 IEROR=10,13
      IF(NEROR(IEROR).EQ.0)GO TO 70
      JEROR=1
      WRITE(24, *)IEROR
2000  FORMAT(/10X,10HDATA ERROR,I5)
70    CONTINUE
C
C***  DECIDE WHEATHER TO CONTINUE OR ECHO DATA.
C
      IF(JEROR EQ.0)RETURN
C
C***  LIST REMAINING DATA
C
      WRITE(24, *)
2010  FORMAT(/10X,14HREMAINING DATA/)
80    CONTINUE
      READ(20,1000)NECHO
1000  FORMAT(150A1)
      WRITE(24,*)ECHO
2020  FORMAT(/10X,80A1)
C      GO TO 80
      type*,'DIAGN2 OVER'
      RETURN
      END
C******************************************************************************
      SUBROUTINE ITERAT(IAXSY,GFLUM,BOUDV,LBOUD,LHEDV,PNORM,EQRHS
     ! ,MELEM,MFRON,MPOIN,MTOTV,NBCON,NDOFM,
     ! NELEM,NEVAB,NGAUS,NITER,NNODL,NNODP,NPOIN,NTOTV,
     ! RELAX,TOLER,XFORC,YFORC)
C*****************************
C_
C_    EXTERNAL SUBROUTINE
C_    PRESCR
C_    FRONTS
C_    WRITER
C_    TOLREL
C_
C*****************************
      COMMON/ARC1/POSGP(3),WEIGP(3)
      COMMON/ARC2/LNODS(105,8),COORD(366,2),NADFM(366),NODFM(366)
      COMMON/ARC3/VARB1(1229),VARB2(1229)
      COMMON/ARC35/EPSN(105,9)
      DIMENSION EQRHS(MTOTV)
      DIMENSION LBOUD(MTOTV),LHEDV(MFRON),PNORM(MFRON),BOUDV(MTOTV)
      DIMENSION GFLUM(MFRON,MFRON)
C
```

```
C
      IITER=0
      DO 1 I = 1,MELEM
      DO 1 J = 1,9
      EPSN(I,J) = 0.0
1     CONTINUE
10    CONTINUE
      IITER=IITER+1
      TYPE*,'IITER=', IITER
      WRITE(35,*)'      IITER=', IITER
      DO 20 ITOTV=1,NTOTV
      EQRHS(ITOTV)=0.0
20    CONTINUE
C
C***  CALL FRONTS TO SET UP AND SOLVE GOVERNING EQUATIONS
C
C67   FORMAT(4X,'YES 4')
      DO 50 I=1,MFRON
      DO 50 J=1,MFRON
      GFLUM(I,J)=0.0
50    CONTINUE
      KOUNT = 1
      TYPE*,KOUNT
      NESPBC = 0
      CALL FRONTS(IAXSY,IITER,MELEM,MFRON,MPOIN,MTOTV,NBCON,NELEM,NEVA
     ! B,NNODL,NNODP,NPOIN,NTOTV,XFORC,YFORC,NGAUS,NESPBC,GFLUM,
     ! BOUDV,LBOUD,LHEDV,PNORM,EQRHS)
      KOUNT = KOUNT + 1
      TYPE*,KOUNT
C***  CALL WRITER TO OUTPUT ITERATION RESULTS.
C
c     CALL WRITER(IITER,MPOIN,MTOTV,NDOFM,NPOIN,MELEM)
C
      KOUNT = KOUNT + 1
      TYPE*,KOUNT
C***  CALL TOLREL TO CHECK CONVERGENCE AND RELAX VALUE IF NOT.
C
      CALL TOLREL(IITER,MTOTV,NCONV,NTOTV,RELAX,TOLER,VARB1,VARB2
     ! ,NADFM,NODFM,MPOIN)
C
      KOUNT = KOUNT + 1
      TYPE*,KOUNT
C***  RETURN TO MASTER NO. IF ITERATIONS EXCEED MAXIMUM.
C
      IF(NCONV EQ.1)GO TO 12
      IF(IITER.LT.NITER)GO TO 10
      WRITE(5,2000)
c     WRITE(35,2000)
2000  FORMAT(//32H SOLUTION HAS FAILED TO CONVERGE)
12    CALL WRITER(IITER,MPOIN,MTOTV,NDOFM,NPOIN,MELEM)
      type*,'ITERAT OVER'
      RETURN
      END
C************************************************************
      SUBROUTINE WRITER(IITER,MPOIN,MTOTV,NDOFM,NPOIN,MELEM)
      COMMON/ARC2/LNODS(105,8),COORD(366,2),NADFM(366),NODFM(366)
      COMMON/ARC3/VARB1(1229),VARB2(1229)
C     OPEN(UNIT=35,FILE='SMALL1.OUT')
      WRITE(5, 111)IITER
111   FORMAT(1X,'IITER=',I5)
      WRITE(5,2000)IITER
      WRITE(35,2000)IITER
2000  FORMAT(//29H RESULTS FOR ITERATION NUMBER,I2,
     ! //14X,4(3HNEW,7X),7X,4(3HOLD,7X)/,
     ! 7H NODE ,2(4X,42HU-VELOCITY PRESSURE V-VELOCITY TEMPERATURE))
```

```
      IODFM=NODFM(IPOIN)
      IADFM=NADFM(IPOIN)
      IF(IODFM.EQ.NDOFM)GO TO 10
      WRITE(5, 2010)IPOIN,(VARB1(IADFM+JODFM-1),JODFM=1,IODFM),
     ! (VARB2(IADFM+JODFM-1),JODFM=1,IODFM)
      WRITE(35, 2010)IPOIN,(VARB1(IADFM+JODFM-1),JODFM=1,IODFM),
     ! (VARB2(IADFM+JODFM-1),JODFM=1,IODFM)
      GO TO 20
10    CONTINUE
      WRITE(5, 2020)IPOIN,(VARB1(IADFM+JODFM-1),JODFM=1,IODFM),
     ! (VARB2(IADFM+JODFM-1),JODFM=1,IODFM)
      WRITE(35, 2020)IPOIN,(VARB1(IADFM+JODFM-1),JODFM=1,IODFM),
     ! (VARB2(IADFM+JODFM-1),JODFM=1,IODFM)

20    CONTINUE
2010  FORMAT(I6,4X,2(E10.3,10X,2E10.3,6X))
2020  FORMAT(I6,4X,2(4E10.3,6X))
      RETURN
      END
C     FINISH*
C*********************************************************
      SUBROUTINE TOLREL(IITER,MTOTV,NCONV,NTOTV,RELAX,TOLER,
     ! VARB1,VARB2,NADFM,NODFM,MPOIN)
      DIMENSION VARB1(MTOTV),VARB2(MTOTV),NADFM(MPOIN),NODFM(MPOIN)
C
C***  CHECK TO SEE IF SOLUTION HAVE CONVERGED.
C
C***  CHECK FOR CONVERGENCE TO REQUIRED TOLERANCE.
C
      NDOFM = 4
      DO 208 IPOIN=1,MPOIN
      IODFM=NODFM(IPOIN)
      IADFM=NADFM(IPOIN)
      IF(IODFM.EQ.NDOFM)GO TO 108
      WRITE(5, 3010)IPOIN,(VARB1(IADFM+JODFM-1),JODFM=1,IODFM),
     ! (VARB2(IADFM+JODFM-1),JODFM=1,IODFM)
      GO TO 208
108   CONTINUE
      WRITE(5, 3020)IPOIN,(VARB1(IADFM+JODFM-1),JODFM=1,IODFM),
     ! (VARB2(IADFM+JODFM-1),JODFM=1,IODFM)
208   CONTINUE
3010  FORMAT(I6,4X,2(E10.3,10X,2E10.3,6X))
3020  FORMAT(I6,4X,2(4E10.3,6X))
      CANLA=0.0
      NCONV=1
      UMAX = VARB1(1)
      PMAX = VARB1(2)
      VMAX = VARB1(3)
      TMAX = VARB1(4)
      DO 10 IPOIN=1,MPOIN
      IODFM = NODFM(IPOIN)
      IADFM = NADFM(IPOIN)
      IF(IODFM.EQ.3) GOTO 16
      IF(ABS(VARB1(IADFM)).GT.ABS(UMAX)) UMAX = VARB1(IADFM)
      IF(ABS(VARB1(IADFM+1)).GT.ABS(PMAX)) PMAX = VARB1(IADFM+1)
      IF(ABS(VARB1(IADFM+2)).GT.ABS(VMAX)) VMAX = VARB1(IADFM+2)
      IF(ABS(VARB1(IADFM+3)).GT.ABS(TMAX)) TMAX = VARB1(IADFM+3)
      GOTO 100
16    IF(ABS(VARB1(IADFM)).GT.ABS(UMAX)) UMAX = VARB1(IADFM)
      IF(ABS(VARB1(IADFM+1)).GT.ABS(VMAX)) VMAX = VARB1(IADFM+1)
      IF(ABS(VARB1(IADFM+2)).GT.ABS(TMAX)) TMAX = VARB1(IADFM+2)
100   CONTINUE
      type*,UMAX,PMAX,VMAX,TMAX
10    CONTINUE
      type*,UMAX,PMAX,VMAX,TMAX
```

```
      PCUT = 0.1*PMAX
      VCUT = 0.1*VMAX
      TCUT = 0.1*TMAX
      type*,UCUT,PCUT,VCUT,TCUT
      DO 18 IPOIN = 1,MPOIN
      IODFM = NODFM(IPOIN)
      IADFM = NADFM(IPOIN)
      ITOTU = IADFM
      IF(IODFM.EQ.3) GOTO 19
      ITOTP = IADFM+1
      ITOTV = ITOTU + 2
      ITOTT = ITOTU + 3
      GOTO 34
19    ITOTV = ITOTU +1
      ITOTT = ITOTU +2
34    IF(ABS(VARB1(ITOTU)).LT.ABS(UCUT)) GOTO 60
      IF(ABS(VARB1(ITOTU)).LT.0.01)GO TO 60
      CANGE=ABS((VARB1(ITOTU)-VARB2(ITOTU))/VARB1(ITOTU))
      IF(CANGE.GT.TOLER)NCONV=0
      IF(CANLA.GT.CANGE)GO TO 60
      CANLA=CANGE
      LTOLA=ITOTU
60    CONTINUE
      IF(IODFM.EQ.3) GOTO 70
      IF(ABS(VARB1(ITOTP)).LT.ABS(PCUT)) GOTO 70
      IF(ABS(VARB1(ITOTP)).LT.0.01)GO TO 70
      CANGE=ABS((VARB1(ITOTP)-VARB2(ITOTP))/VARB1(ITOTP))
      IF(CANGE.GT.TOLER)NCONV=0
      IF(CANLA.GT.CANGE)GO TO 70
      CANLA=CANGE
      LTOLA=ITOTP
70    CONTINUE
      IF(ABS(VARB1(ITOTV)).LT.ABS(VCUT)) GOTO 80
      IF(ABS(VARB1(ITOTV)).LT.0.01)GO TO 80
      CANGE=ABS((VARB1(ITOTV)-VARB2(ITOTV))/VARB1(ITOTV))
      IF(CANGE.GT.TOLER)NCONV=0
      IF(CANLA.GT.CANGE)GO TO 80
      CANLA=CANGE
      LTOLA=ITOTV
80    CONTINUE
      IF(ABS(VARB1(ITOTT)).LT.ABS(TCUT)) GOTO 18
      IF(ABS(VARB1(ITOTT)).LT.0.01)GO TO 18
      CANGE=ABS((VARB1(ITOTT)-VARB2(ITOTT))/VARB1(ITOTT))
      IF(CANGE.GT.TOLER)NCONV=0
      IF(CANLA.GT.CANGE)GO TO 18
      CANLA=CANGE
      LTOLA=ITOTT
18    CONTINUE
      type*,'5000'
      WRITE(5,2000)LTOLA,CANLA
      WRITE(35,2000)LTOLA,CANLA
2000  FORMAT(/37X,' LARGEST CHANGE OCCURE ATDOF NO ',I5,/,
     ! 10X,' CHANGE  =',F10.4)
      IF(NCONV.EQ.0)GO TO 20
      type*,'6000'
      WRITE(5,2010) IITER
      WRITE(35,2010) IITER
2010  FORMAT(//45H SOLUTION HAS CONVERGED TO REQUIRED TOLERANCE/,
     ! 3x,'IN',3x,I5,3x,'ITERATIONS')
      type*,'TOLREL OVER'
      RETURN
20    CONTINUE
      IF(IITER.EQ.1)GO TO 40
C
C***  RELAX VARIABLES FOR NEXT ITERATIONS
```

```
      type*,'7000'
      DO 222 IPOIN=1,MPOIN
      IODFM=NODFM(IPOIN)
      IADFM=NADFM(IPOIN)
      DO 21 JODFM=1,IODFM
      TEMPY1=VARB1(IADFM+JODFM-1)
      TEMPY2=VARB2(IADFM+JODFM-1)
      VARB2(IADFM+JODFM-1)=TEMPY2+RELAX*(TEMPY1-TEMPY2)
21    CONTINUE
222   CONTINUE
      GOTO 225
40    CONTINUE
      DO 50 ITOTV=1,NTOTV
      VARB2(ITOTV)=VARB1(ITOTV)
50    CONTINUE
225   CONTINUE
      RETURN
      END
C*****************************************************************
      SUBROUTINE FRONTS(IAXSY,IITER,MELEM,MFRON,MPOIN,MTOTV,NBCON,NE
     ! LEM,NEVAB,NNODL,NNODP,NPOIN,NTOTV,XFORC,YFORC,NGAUS,NESPBC,GFLU
     ! M,BOUDV,LBOUD,LHEDV,PNORM,EQRHS)
C*************************************
C_
C_    EXTERNAL SUBROUTINES
C_    MATRIX
C_
C*************************************
      DIMENSION LBOUD(MTOTV),LHEDV(MFRON),PNORM(MFRON),BOUDV(MTOTV)
      DIMENSION EQRHS(MTOTV)
      COMMON/ARC2/LNODS(105,8),COORD(366,2),NADFM(366),NODFM(366)
      COMMON/ARC3/VARB1(1229),VARB2(1229)
      DIMENSION LOCEL(28),NDEST(28),FLUMX(28,28),GFLUM(MFRON,MFRON)
      OPEN(UNIT=25,FORM='UNFORMATTED',FILE='TTRASH.DAT')
      IF(IITER.GT.1)GO TO 40
C
C***  ON FIRST ITERATION ONLY FIND LAST APPEARANCE OF EACH NOD.
C
      DO 30 IPOIN=1,NPOIN
      LASTE=0
      DO 20 IELEM=1,NELEM
      DO 10 INODP=1,NNODP
      IF(LNODS(IELEM,INODP).NE.IPOIN)GO TO 10
      LASTE=IELEM
      LASTN=INODP
      GO TO 20
10    CONTINUE
20    CONTINUE
      LNODS(LASTE,LASTN)=-IPOIN
30    CONTINUE
40    CONTINUE
C
C***  INITIALISE HEADING AND GRAND FLUID MATRIX.
C
C      REWIND 25
C     REWIND 10
      NCRIT=MFRON-NEVAB
      NFRON=0
      DO 50 IFRON = 1,MFRON
      DO 50 JFRON = 1,MFRON
      GFLUM(IFRON,JFRON) = 0.0
50    CONTINUE
      KELEM=0
C
```

```
C***    START ASSEMBLY BY FORMING ELEMENTAL MATRICES
C
60      CONTINUE
        KELEM=KELEM+1
        CALL MATRIX(IAXSY,MELEM,EQRHS,FLUMX,MPOIN
     !  ,MTOTV,NEVAB,NNODL,NNODP,XFORC,YFORC
     !  ,KELEM,NELEM,NGAUS,IITER)
        KEVAB=0
C
C***    CREAT GLOBAL DOF ARRAY FOR EACH LOCAL ELEMENT DOF.
C
        DO 70 INODP=1,NNODP
        KPOIN=LNODS(KELEM,INODP)
        IADFM=NADFM(IABS(KPOIN))
        LODFM=NODFM(IABS(KPOIN))
        DO 70 IODFM=1,LODFM
        KEVAB=KEVAB+1
        LOCEL(KEVAB)=IADFM+IODFM-1
        IF(KPOIN.LT.0)LOCEL(KEVAB)=-LOCEL(KEVAB)
70      CONTINUE
C
C***    FIT EACH DOF INTO THE FRONT WIDTH EXTENDING IF NECESSARY.
C
        DO 120 IEVAB=1,NEVAB
        KTOTV=LOCEL(IEVAB)
        IF(NFRON.EQ.0)GO TO 90
        DO 80 IFRON=1,NFRON
        KFRON=IFRON
        IF(IABS(KTOTV).EQ.IABS(LHEDV(KFRON)))GO TO 110
80      CONTINUE
90      CONTINUE
        NFRON=NFRON+1
        IF(NFRON.LE.MFRON)GO TO 100
        WRITE(5, 2000)
2000    FORMAT(//3X,'PROGRAM HALTED FRONTWIDTH IS TOO SMALL')
        STOP
100     CONTINUE
        NDEST(IEVAB)=NFRON
        LHEDV(NFRON)=KTOTV
        GO TO 120
110     CONTINUE
        NDEST(IEVAB)=KFRON
        LHEDV(KFRON)=KTOTV
120     CONTINUE
C
C***    ASSEMBLE NEW ELEMENT INTO GRAND FLUID MATRIX.
C
        DO 130 IEVAB=1,NEVAB
        IFRON=NDEST(IEVAB)
        DO 130 JEVAB=1,NEVAB
        JFRON=NDEST(JEVAB)
        GFLUM(JFRON,IFRON)=GFLUM(JFRON,IFRON)+FLUMX(JEVAB,IEVAB)
130     CONTINUE
        IF(NFRON.LT.NCRIT.AND.KELEM.LT.NELEM)GO TO 60
140     CONTINUE
        NFSUM=0
        PIVOT=0.0
C
C***    CHECK LAST APPEARANCE OF EACH DOF PROCESS BOUNDRY CONDITIONS.
C
        DO 170 IFRON=1,NFRON
        IF(LHEDV(IFRON).GE.0)GO TO 170
        NFSUM=1
        IF(LBOUD(IABS(LHEDV(IFRON))).NE.1)GO TO 160
        KTOTV=IABS(LHEDV(IFRON))
        LBOUD(KTOTV)=-1
```

```
      DO 150 LFRON=1,NFRON
      GFLUM(IFRON,LFRON)=0.0
150   CONTINUE
      GFLUM(IFRON,IFRON)=1.0
160   CONTINUE
C
C***  SEARCH FOR LARGEST PIVOTAL VALUE
C
      PIVOG=GFLUM(IFRON,IFRON)
      IF(ABS(PIVOG).LT.ABS(PIVOT))GO TO 170
      PIVOT=PIVOG
      LPIVT=IFRON
170   CONTINUE
      IF(NFSUM.EQ.0)GO TO 60
      KTOTV=IABS(LHEDV(LPIVT))
C     REDUCED THE VALUE 1E-7 TO 1E-10
      IF(ABS(PIVOT).GT.1.0E-12)GO TO 180
      TYPE*,PIVOT
      WRITE(5, 2010)KTOTV,PIVOT
2010  FORMAT(//32H PROGRAM HALTED ILL-CONDITIONING,//17H D.O.FREEDOM
     !      ,I4,/13H PIVOT VALUE   ,E9.2)
      STOP
180   CONTINUE
C
C***  NORMALISE PIVOTAL EQUQTION
C
      DO 190 IFRON=1,NFRON
      PNORM(IFRON)=GFLUM(LPIVT,IFRON)/PIVOT
190   CONTINUE
      RHSID=EQRHS(KTOTV)/PIVOT
      EQRHS(KTOTV)=RHSID
C
C***  ELEMINATION OF PIVOTAL EQUATION REDUCING FRONT WIDTH.
C
      IF(LPIVT.EQ.1)GO TO 250
      DO 240 IFRON=1,LPIVT-1
      FACOR=GFLUM(IFRON,LPIVT)
      IF(FACOR.EQ.0)GO TO 210
      DO 200 JFRON=1,LPIVT-1
      GFLUM(IFRON,JFRON)=GFLUM(IFRON,JFRON)-FACOR*PNORM(JFRON)
200   CONTINUE
210   CONTINUE
      IF(LPIVT.EQ.NFRON)GO TO 230
      DO 220 JFRON=LPIVT+1,NFRON
      GFLUM(IFRON,JFRON-1)=GFLUM(IFRON,JFRON)-FACOR*PNORM(JFRON)
220   CONTINUE
230   CONTINUE
      ITOTV=IABS(LHEDV(IFRON))
      EQRHS(ITOTV)=EQRHS(ITOTV)-FACOR*RHSID
240   CONTINUE
250   CONTINUE
      IF(LPIVT.EQ.NFRON)GO TO 300
      DO 290 IFRON=LPIVT+1,NFRON
      FACOR=GFLUM(IFRON,LPIVT)
      IF(LPIVT.EQ.1)GO TO 270
      DO 260 JFRON=1,LPIVT-1
      GFLUM(IFRON-1,JFRON)=GFLUM(IFRON,JFRON)-FACOR*PNORM(JFRON)
260   CONTINUE
270   CONTINUE
      DO 280 JFRON=LPIVT+1,NFRON
      GFLUM(IFRON-1,JFRON-1)=GFLUM(IFRON,JFRON)-FACOR*PNORM(JFRON)
280   CONTINUE
      ITOTV=IABS(LHEDV(IFRON))
      EQRHS(ITOTV)=EQRHS(ITOTV)-FACOR*RHSID
290   CONTINUE
```

```
300     CONTINUE
C
C***    WRITE OUT NONFIXED PIVOTAL EQUATION ON TAPE.
C
        IF(LBOUD(KTOTV).NE.0)GO  TO 310
        WRITE(25)NFRON,LPIVT,(LHEDV(IFRON),PNORM(IFRON),IFRON=1,NFRON)
310     CONTINUE
        DO 320 IFRON=1,NFRON
        GFLUM(IFRON,NFRON)=0.0
        GFLUM(NFRON,IFRON)=0.0
320     CONTINUE
        IF(LPIVT EQ NFRON)GO TO 340
        DO 330 IFRON=LPIVT,NFRON-1
        LHEDV(IFRON)=LHEDV(IFRON+1)
330     CONTINUE
340     CONTINUE
        NFRON=NFRON-1
C
C***    ASSEMBLE ELIMINATE OR BACK SUBSTITUTE.
C
        IF(NFRON.GT.NCRIT)GO TO 140
        IF(KELEM.LT.NELEM)GO TO 60
        IF(NFRON.GT 0)GO TO 140
C
CC***   BACK SUBSTITUTION
C
        DO 350 ITOTV=1,NTOTV
        VARB1(ITOTV)=BOUDV(ITOTV)
        LBOUD(ITOTV)=-LBOUD(ITOTV)
350     CONTINUE
        DO 370 ITOTV=1,NTOTV-NBCON
        BACKSPACE 25
        READ(25)NFRON,LPIVT,(LHEDV(IFRON),PNORM(IFRON),IFRON=1,NFRON)
        KTOTV=IABS(LHEDV(LPIVT))
        TEMPR=0.0
        PNORM(LPIVT)=0.0
        DO 360 IFRON=1,NFRON
        TEMPR=TEMPR-PNORM(IFRON)*VARB1(IABS(LHEDV(IFRON)))
360     CONTINUE
        VARB1(KTOTV)=EQRHS(KTOTV)+TEMPR
        BACKSPACE 25
370     CONTINUE
        type*,'FFRONTS OVER'
        RETURN
        END
C*************************************************************************
C*********************************************************************
        SUBROUTINE MATRIX(IAXSY,MELEM,EQRHS,FLUMX,
     !  MPOIN,MTOTV,NEVAB,NNODL,NNODP,XFORC
     !  ,YFORC,KELEM,NELEM,NGAUS,IITER)
C       ***INSERTED THE DUMMY VARIABLE "NSIDE" ABOVE********
        DIMENSION CARTL(2,4),CARTP(2,8),ERHSU(8),ERHSV(8),FLUMX(28,28)
        DIMENSION ERHST(8),VISC(9)
        DIMENSION SHAPL(4),SHAPP(8),AREAW(9)
        COMMON/ARC2/LNODS(105,8),COORD(366,2),NADFM(366),NODFM(366)
        DIMENSION EQRHS(MTOTV)
        COMMON/ARC3/VARB1(1229),VARB2(1229)
        COMMON/AREA3/CARPG(2,72),SHAPG(72),CARLG(2,36),SHALG(36)
C       COMMON/ARC4/AREAW(9)
        COMMON/ARC10/CUTVEL,DCUT,BSTAR
        COMMON/ARC12/HSTAR,TFSTAR,FRIFAC,PI,EN,COSA,SINA,CONRAT
        COMMON/ARC13/USTAR,PECLET,SIGMA,RHO,MBOUEL
        COMMON/ARC14/ IBEL(50),ISURF(50),ISIDE(50)
        COMMON/ARC35/EPSN(105,9)
C        OPEN(UNIT=28,DEVICE='DSK',FILE='TRASH.OUT')
        type*,(LNODS(KELEM,IJK),IJK=1,8)
```

```
      CALL DRIVES(COORD,LNODS,MELEM,MPOIN,NELEM,3,NNODL,NNODP,
     ! KELEM,AREAW)
C
C***  INITIALISE ARRAYS
C
      DO 10 INODP=1,NNODP
      ERHSU(INODP)=0.0
      ERHSV(INODP)=0.0
      ERHST(INODP)=0 0
10    CONTINUE
      DO 20 IEVAB=1,NEVAB
      DO 20 JEVAB=1,NEVAB
      FLUMX(IEVAB,JEVAB)=0.0
20    CONTINUE
C****************************************
C***  LOOP TO CARRY OUT GAUSS INTEGRATIONS
C
      DO 100 IGAUS=1,9
      KIGS=IGAUS
      DAREA=AREAW(KIGS)
      DO 30 INODP=1,NNODP
      SHAPP(INODP)=SHAPG(NNODP*(IGAUS-1)+INODP)
      DO 30 IDIME=1,2
      CARTP(IDIME.INODP)= CARPG(IDIME,NNODP*(IGAUS-1)+INODP)
30    CONTINUE
      DO 40 INODL=1,NNODL
      SHAPL(INODL)=SHALG(NNODL*(IGAUS-1)+INODL)
      DO 40 IDIME=1,2
      CARTL(IDIME,INODL)=CARLG(IDIME,NNODL*(IGAUS-1)+INODL)
40    CONTINUE
      IELEM=KELEM
      UVELY=0.0
      RADUS=0.0
      VVELY=0.0
C
C***  EVALUATE RADIUS AND PREVIOUS VELOCITIES AT GAUSS POINTS
C
      DUDX = 0.0
      DUDY = 0.0
      DVDX  = 0.0
      DVDY  = 0.0
      DO 50 INODP=1,NNODP
      KPOIN=IABS(LNODS(KELEM,INODP))
      ITOTU=NADFM(KPOIN)
      ITOTV=ITOTU+NODFM(KPOIN)-2
      SHAPE=SHAPP(INODP)
      UVELY=UVELY+VARB2(ITOTU)*SHAPE
      RADUS=RADUS+COORD(KPOIN,2)*SHAPE
      VVELY=VVELY+VARB2(ITOTV)*SHAPE
      CARXI=CARTP(1,INODP)
      CARYI=CARTP(2,INODP)
      UVEL = VARB2(ITOTU)
      VVEL = VARB2(ITOTV)
      DUDX = DUDX + CARXI*UVEL
      DUDY = DUDY + CARYI*UVEL
      DVDX = DVDX + CARXI*VVEL
      DVDY = DVDY + CARYI*VVEL
50    CONTINUE
      EPS = SQRT(2.0*(DUDX**2+0.5*((DUDY+DVDX)**2)+DVDY**2))
      IF(EPS.LT.0.01) EPS = 0.01
      EPSN(KELEM,KIGS)= EPS
      VISC(KIGS) = (1.0+BSTAR*EPS**(1.0/EN))/(1.732*EPS)
      IF(IAXSY.EQ.1)DAREA=DAREA*RADUS
C
C***  PUT HEAT GEN. TERM &/or BODY FORCES INTO LOCAL RHS VECTOR.
```

```
C
      DO 60 INODP=1,NNODP
      ERHST(INODP) = ERHST(INODP)+SHAPP(INODP)*EPS*PECLET*DAREA
     ! *EPS*VISC(KIGS)
60    CONTINUE
      DO 90 ICON1=1,4
      DO 90 ICON2=1,2
      IROWU=(ICON1-1)*7+4*ICON2-3
      IROWV=IROWU+3-ICON2
      IROWT = IROWU+4-ICON2
      INODP=2*(ICON1-1)+ICON2
      SHAPI=SHAPP(INODP)
      CARXI=CARTP(1,INODP)
      CARYI=CARTP(2,INODP)
      IF(ICON2 EQ.1) IROWP=IROWU+1
      DO 80 JCON1=1,4
      JCOLP=(JCON1-1)*7+2
      KGALI = (KIGS-1)*NNODL+JCON1
C
C***  PUT PRESSURE TERM IN MOMENTUM EQUATIONS.
C
      FLUMX(IROWU,JCOLP)=FLUMX(IROWU,JCOLP)-CARXI*SHALG(KGALI)
     ! *DAREA
      FLUMX(IROWV,JCOLP)=FLUMX(IROWV,JCOLP)-CARYI*SHALG(KGALI)*
     ! DAREA
      DO 80 JCON2=1,2
      JCOLU=(JCON1-1)*7+4*JCON2-3
      JCOLV=JCOLU+3-JCON2
      JCOLT = JCOLU + 4-JCON2
      JNODP=2*(JCON1-1)+JCON2
      SHAPJ=SHAPP(JNODP)
      CARXJ=CARTP(1,JNODP)
      CARYJ=CARTP(2,JNODP)
C
C***  PUT DIFFUSION AND CONVECTION TERMS IN MOMENTUM EQUATIONS.
C
C*******************************************************************
      DIFFU=(CARXI*2.0*CARXJ+CARYI*CARYJ)*VISC(KIGS)*DAREA
      CONVC=((UVELY*CARXJ+VVELY*CARYJ)*(SHAPI*DAREA))/SIGMA
      FLUMX(IROWU,JCOLU)=FLUMX(IROWU,JCOLU)+DIFFU+CONVC
      DIFFU = (CARXI*CARXJ+2.0*CARYI*CARYJ)*VISC(KIGS)*DAREA
      FLUMX(IROWV,JCOLV)=FLUMX(IROWV,JCOLV)+DIFFU+CONVC
      FLUMX(IROWU,JCOLV) = CARYI*CARXJ*VISC(KIGS)*DAREA
     ! +FLUMX(IROWU,JCOLV)
      FLUMX(IROWV,JCOLU) =FLUMX(IROWV,JCOLU)+ CARXI*CARYJ*VISC(KIGS)
     ! *DAREA
C
C*    FORM ENERGY EQUATION
C
C***  PUT CONDUCTION AND CONVECTION TERMS IN THE ENERGY EQUATION
C
      ENCOND = (CARXI*CARXJ+CARYI*CARYJ)*DAREA
      ENCONV = (UVELY*CARXJ+VVELY*CARYJ)*SHAPI*DAREA*PECLET
      FLUMX(IROWT,JCOLT) = FLUMX(IROWT,JCOLT)+ENCOND+ENCONV
      IF(ICON2.EQ.2)GO TO 70
C
C***  FORM CONTINUITY EQUATION.
C
     FLUMX(IROWP,JCOLU)=FLUMX(IROWP,JCOLU)+SHAPL(ICON1)*DAREA*CARXJ
     FLUMX(IROWP,JCOLV)=FLUMX(IROWP,JCOLV)+SHAPL(ICON1)*DAREA*CARYJ
      FLUMX(IROWP,JCOLP)=0.0
      FLUMX(IROWP,JCOLT)=0.0
70    CONTINUE
80    CONTINUE
90    CONTINUE
100   CONTINUE
```

```
C***     TO GLOBAL ARRAY.
C
         DO 110 INODP=1,NNODP
         KINP=INODP
         KPOIN=IABS(LNODS(IELEM,KINP))
         ITOTU=NADFM(KPOIN)
         ITOTV=ITOTU+NODFM(KPOIN)-2
         ITOTT = ITOTU+NODFM(KPOIN)-1
         EQRHS(ITOTU)=EQRHS(ITOTU)+ERHSU(INODP)
         EQRHS(ITOTV)=EQRHS(ITOTV)+ERHSV(INODP)
         EQRHS(ITOTT) = EQRHS(ITOTT) + ERHST(INODP)
110      CONTINUE
         DO 120 I=1,MBOUEL
         JK = I
         IF(KELEM.NE IBEL(JK)) GOTO 120
         KBEL = IBEL(JK)
         KSURF = ISURF(JK)
         IF((KSURF.LE 3).OR.(KSURF.EQ.7)) GOTO 120
199      KSIDE = ISIDE(JK)
         TYPE*,KSURF,KBEL,KSIDE
         type*,'CALLING BOUCON'
         CALL BOUCON(KBEL,KSURF,KSIDE,FLUMX,EQRHS,VARB2,MELEM,
     1   MPOIN,NNODP,MTOTV,IITER)
120      CONTINUE
1987     TYPE 1988
1988     FORMAT(3X,'MATRIX OVER')
         RETURN
         END
C******************************************************************************
         SUBROUTINE DRIVES(COORD,LNODS,MELEM,MPOIN,NELEM,NGAUS,
     !   NNODL,NNODP,IELEM,AREAW)
C***************************
C-
C-       EXTERNAL SUBROUTINES
C-       DJACOB
C-
C***************************
         DIMENSION COORD(MPOIN,2),LNODS(MELEM,8)
         DIMENSION DERIV(2,8),DJACI(2,2),DJACK(2,2),SHAPE(8)
         DIMENSION CARTP(2,8),AREAW(9)
         COMMON/ARC1/POSGP(3),WEIGP(3)
         COMMON/AREA3/CARPG(2,72),SHAPG(72),CARLG(2,36),SHALG(36)
C        COMMON/AREA4/AREAW(9)
C
C***     REWIND TAPE BEFORE WRITING ON SHAPE FUNCTIONS
C
C        REWIND 10
C
C***     SET UP POSITIONS AND WEIGHTS FOR 3 POINT GUASS RULE
C
         POSGP(1)=0.7745966692
         POSGP(2)=0.0
         POSGP(3)=-POSGP(1)
         WEIGP(1)=0.5555555556
         WEIGP(2)=0.8888888889
         WEIGP(3)=WEIGP(1)
C
C***     CALCULATE SHAPE FUNCTIONS AND DERIVETIVES FOR ELEMENTS
C
         LGAUS=0
         DO 50 IGAUS=1,NGAUS
         DO 50 JGAUS=1,NGAUS
         LGAUS=LGAUS+1
         XEQIV=POSGP(IGAUS)
         YEQIV=POSGP(JGAUS)
C
```

```
C
      CALL SHAPE8(DERIV,SHAPE,XEQIV,YEQIV)
C
C***  SET UP JACOBIAN MATRIX AND INVERSE
C
      CALL DJACOB(DERIV,DETJB,DJACI,DJACK,IELEM
     ! ,MELEM,MPOIN,NNODP,LGAUS)
C
C***  CALCULATE GLOBAL DERIVETIVES AND AREA*GAUSS WEIGHTS
C
      DO 10 IDIME=1,2
      DO 10 INODP=1,NNODP
      CARTP(IDIME,INODP)=0.0
      DO 10 JDIME=1,2
      CARTP(IDIME,INODP)=CARTP(IDIME,INODP)+DJACI(IDIME,JDIME)
     ! *DERIV(JDIME,INODP)
10    CONTINUE
      AREAW(LGAUS)=DETJB*WEIGP(IGAUS)*WEIGP(JGAUS)
C
C***  PUT SHAPE FUNCTIONS AND DERIVETIVES IN ELEMENT MATRIX
C
      DO 20 INODP=1,NNODP
      KAPA=(LGAUS-1)*NNODP+INODP
      SHAPG(KAPA)=SHAPE(INODP)
      DO 20 IDIME=1,2
      CARPG(IDIME,KAPA)=CARTP(IDIME,INODP)
20    CONTINUE
C
C***  USE GAUSS POSITIONS TO CALCULATE LOCAL VALUES
C
      CALL SHAPE4(DERIV,SHAPE,XEQIV,YEQIV)
C
C***  CALCULATE GLOBAL DERIVATIVES FOR LINEAR FUNCTIONS
C
      DO 30 IDIME=1,2
      DO 30 INODL=1,NNODL
      CARTP(IDIME,INODL)=0.0
      DO 30 JDIME=1,2
      CARTP(IDIME,INODL)=CARTP(IDIME,INODL)+
     ! DJACI(IDIME,JDIME)*DERIV(JDIME,INODL)
30    CONTINUE
C
C***  PUT SHAPE FUNCTIONS AND DERIVATIVES IN ELEMENT MATRIX
C
      DO 40 INODL=1,NNODL
      KGALI=(LGAUS-1)*NNODL+INODL
      SHALG(KGALI)=SHAPE(INODL)
      DO 40 IDIME=1,2
      CARLG(IDIME,KGALI)=CARTP(IDIME,INODL)
40    CONTINUE
50    CONTINUE
      RETURN
      END
C*******************************************************
      SUBROUTINE SHAPE8(DERIV,SHAPE,XEQIV,YEQIV)
      DIMENSION DERIV(2,8),SHAPE(8)
C
C***  PARABOLIC ELEMENT ANTICLOCKWISE NOD NUMBERING
C
      X=XEQIV
      Y=YEQIV
      XY=X*Y
      XX=X*X
      YY=Y*Y
      XXY=XX*Y
```

```
      X2=X*2.
      Y2=Y*2.
      XY2=XY*2
      SHAPE(1)=(-1.+XY+XX+YY-XXY-XYY)*.25
      SHAPE(2)=(1.-Y-XX+XXY)*.5
      SHAPE(3)=(-1.-XY+XX+YY-XXY+XYY)*.25
      SHAPE(4)=(1.+X-YY-XYY)*.5
      SHAPE(5)=(-1.+XY+XX+YY+XXY+XYY)*.25
      SHAPE(6)=(1.+Y-XX-XXY)*.5
      SHAPE(7)=(-1.-XY+XX+YY+XXY-XYY)*.25
      SHAPE(8)=(1.-X-YY+XYY)*.5
      DERIV(1,1)=(Y+X2-XY2-YY)*.25
      DERIV(1,2)=-X+XY
      DERIV(1,3)=(-Y+X2-XY2+YY)*.25
      DERIV(1,4)=(1.-YY)*.5
      DERIV(1,5)=(Y+X2+XY2+YY)*.25
      DERIV(1,6)=-X-XY
      DERIV(1,7)=(-Y+X2+XY2-YY)*.25
      DERIV(1,8)=(-1.+YY)*.5
      DERIV(2,1)=(X+Y2-XX-XY2)*.25
C
      DERIV(2,2)=(-1.+XX)*.5
      DERIV(2,3)=(-X+Y2-XX+XY2)*.25
      DERIV(2,4)=-Y-XY
      DERIV(2,5)=(X+Y2+XX+XY2)*.25
      DERIV(2,6)=(1.-XX)*.5
      DERIV(2,7)=(-X+Y2+XX-XY2)*.25
      DERIV(2,8)=-Y+XY
      RETURN
      END
C*****************************************************************
      SUBROUTINE DJACOB(DERIV,DETJB,DJACI,DJACK,IELEM
     ! ,MELEM,MPOIN,NNODP,LGAUS)
      DIMENSION DERIV(2,8),DJACI(2,2),DJACK(2,2)
      COMMON/ARC2/LNODS(105,8),COORD(366,2),NADFM(366),NODFM(366)
C
C***  SET UP TEMPORARY MATRIX TO ALLOW THE JACOBIAN TO BE FORMED
C
      DO 20 IDIME=1,2
      DO 20 JDIME=1,2
      TEMPY=0.0
      DO 10 INODP=1,NNODP
      KPOIN=IABS(LNODS(IELEM,INODP))
      KAG=(LGAUS-1)*NNODP+INODP
      TEMPY=TEMPY+DERIV(IDIME,INODP)*COORD(KPOIN,JDIME)
10    CONTINUE
      DJACK(IDIME,JDIME)=TEMPY
20    CONTINUE
      DETJB=DJACK(1,1)*DJACK(2,2)-DJACK(1,2)*DJACK(2,1)
C2020  FORMAT(5X,'DETERMENT VALUE=',F15.8)
C
C***  CHECK FOR NEGATIVE OR ZERO DETERMINENT
C

      IF(DETJB)30,30,40
30    CONTINUE
      WRITE(5,2000)IELEM,LGAUS
      STOP
40    CONTINUE
C
C***  INVERT TEMPORARY MATRIX TO FORM JACOBIAN
C
      DJACI(1,1)=DJACK(2,2)/DETJB
      DJACI(2,2)=DJACK(1,1)/DETJB
      DJACI(1,2)=-DJACK(1,2)/DETJB
```

```
      DJACI(2,1)=-CJACK(2,1)/DETJB
2000  FORMAT(//37H NON POSITIVE DETERMINENT FOR ELEMENT,2I4)
      RETURN
      END
C*****************************************************************
      SUBROUTINE SHAPE4(DERIV,SHAPE,XEQIV,YEQIV)
      DIMENSION DERIV(2,4),SHAPE(4)
C
C******* LINEAR ELEMENT ANTICLOCKWISE NODNUMBERING
C
      X=XEQIV
      Y=YEQIV
      XY=X*Y
      SHAPE(1)=(1.-X-Y+XY)*.25
      SHAPE(2)=(1.+X-Y-XY)*.25
      SHAPE(3)=(1.+X+Y+XY)*.25
      SHAPE(4)=(1.-X+Y-XY)*.25
      DERIV(1,1)=(-1.+Y)*.25
      DERIV(1,2)=(1.-Y)*.25
      DERIV(1,3)=(1.+Y)*.25
      DERIV(1,4)=(-1.-Y)*.25
      DERIV(2,1)=(-1.+X)*.25
      DERIV(2,2)=(-1.-X)*.25
      DERIV(2,3)=(1.+X)*.25
      DERIV(2,4)=(1.-X)*.25
      RETURN
      END
C**************************************************************
      SUBROUTINE BOUCON(KBEL,KSURF,KSIDE,FLUMX,EQRHS,VARB2,
     1 MELEM,MPOIN,NNODP,MTOTV,IITER)
      COMMON/ARC1/POSGP(3),WEIGP(3)
      COMMON/ARC2/LNODS(105,8),COORD(366,2),NADFM(366),NODFM(366)
      COMMON/ARC12/HSTAR,TFSTAR,FRIFAC,PI,EN,COSA,SINA,CONRAT
      COMMON/ARC13/USTAR,PECLET,SIGMA,RHO,MBOUEL
      COMMON/ARC30/ANGLN(366)
      DIMENSION EQRHS(MTOTV),VARB2(MTOTV),FLUMX(28,28)
      DIMENSION M(3),NODE(3),DERX(2,8),SHAP1(8),SHAPN(8)
      DIMENSION RHSU(8),RHSV(8),RHST(8),CART(2,8)
      DIMENSION CJACK(2,2),CJACI(2,2),COSN(3),SINN(3)
C
C**   IDENTIFYING LOCAL NODE NOS.
      IF(KSURF.LE.3) GOTO 88
      M(1)= 1
      M(2) = 5
      M(3) = 8
      NODE(1) = 2*KSIDE - 1
      NODE(2) = NODE(1) + 1
      NODE(3) = NODE(2) + 1
      IF(NODE(3).GT.8) NODE(3) = NODE(3)-8
      DO 145 I = 1,8
      RHSU(I) = 0.0
      RHSV(I) = 0.0
      RHST(I) = 0.0
145   CONTINUE
C-----------------------------------------------------
C      GAUSS POINT EVALUATION STARTS
C-----------------------------------------------------
      DO 150 IGAUS = 1,3
      GOTO (10,20,30,40) , KSIDE
C-----------------------------------------------------
C     CALCULATING LENGTH OF THE ELEMENT AT EACH GAUSS PT.
C-----------------------------------------------------------
10    CONTINUE
      XEQIV = POSGP(IGAUS)
      YEQIV = -1.0
      ITEMP = 1
```

```
20        CONTINUE
          XEQIV = 1.0
          YEQIV = POSGP(IGAUS)
          ITEMP = 2
          GOTO 50
30        CONTINUE
          XEQIV = POSGP(IGAUS)
          YEQIV = 1.0
          ITEMP = 1
          GOTO 50
40        CONTINUE
          XEQIV = -1.0
          YEQIV = POSGP(IGAUS)
          ITEMP = 2
50        CONTINUE
          CALL SHAPE8(DERX,SHAP1,XEQIV,YEQIV)
          CALL DJACOB(DERX,DETJB,CJACI,CJACK,KBEL,MELEM
     1    ,MPOIN,NNODP,IGAUS)
          TEMPY = SQRT(CJACK(ITEMP,1)**2+CJACK(ITEMP,2)**2)
          SLETH = TEMPY*WEIGP(IGAUS)
C-----------------------------------------------------------------------
C CALCULATING VISCOSITY & STRAIN RATE AT GAUSS POINT FOR 5th SURF. ONLY.
C_----------------------------------------------------------------------
          IF(KSURF NE 5) GOTO 15
          DO 51 IDIME = 1,2
          DO 51 INODP = 1,NNODP
          CART(IDIME,INODP)=0.0
          DO 51 JDIME = 1,2
          CART(IDIME,INODP)=CART(IDIME,INODP)+DERX(JDIME,INODP)
     1    *CJACI(IDIME,JDIME)
51        CONTINUE
          UVELY = 0.0
          VVELY = 0.0
          DUDX = 0 0
          DUDY = 0.0
          DVDX = 0 0
          DVDY = 0.0
          DO 60 I = 1,8
          KPOIN = IABS(LNODS(KBEL,I))
          ITOTU =NADFM(KPOIN)
          ITOTV = ITOTU + NODFM(KPOIN) - 2
          SHAP = SHAP1(I)
          UVELY = UVELY + VARB2(ITOTU)*SHAP
          VVELY = VVELY + VARB2(ITOTV)*SHAP
          CARXI = CART(1,I)
          CARYI = CART(2,I)
          UVEL = VARB2(ITOTU)
          VVEL = VARB2(ITOTV)
          DUDX = DUDX + CARXI*UVEL
          DUDY = DUDY + CARYI*UVEL
          DVDX = DVDX + CARXI*VVEL
          DVDY = DVDY + CARYI*VVEL
60        CONTINUE
          EPS = SQRT(2.0*(DUDX**2+0.5*(DUDY+DVDX)**2+DVDY**2))
          IF(EPS.LT.0.01) EPS = 0.01
          VISC = (1.0+BSTAR*EPS**(1.0/EN))/(1.732*EPS)
          VT = VVELY*COSA + UVELY*SINA
C-----------------------------------------------------------------
C         EVALUATING GAUSS PT. INTEGRALS FOR LOCAL RHS VECTOR
C-----------------------------------------------------------------
          DO 13 I = 1,3
          NOD = NODE(I)
          SHAP = SHAP1(NOD)
          FACTOR = FRIFAC*VISC*EPS
          RHSU(NOD) = RHSU(NOD) - SHAP*FACTOR*SINA*SLETH
```

```
      RHST(NOD) = RHST(NOD) + SHAP*PECLET*FACTOR*VT*CONRAT*SLETH
      SHEAR = SHAP*PECLET*FACTOR*CONRAT*SLETH
13    CONTINUE
C--------------------------------------------------------
C   SKIPPING CONVECTIVE BC. FOR 5th SURFACE
C--------------------------------------------------------
      IF(KSURF.EQ.5) GOTO 150
15    CONTINUE
C--------------------------------------------------------
C   INCORPORATING APPROPRIATE CONVECTIVE HT.TR. BC.
C--------------------------------------------------------
      DO 75 II = 1,3
      INODP = 2*(KSIDE-1) +II
      IROWU = 7*(KSIDE-1) + M(II)
      IF(INODP.GT.8) INODP = INODP - 8
      IF(IROWU.GT.28) IROWU = IROWU - 28
      KPOIN = IABS(LNODS(KBEL,INODP))
      IROWT = IROWU + NODFM(KPOIN) -1
      SHAPI = SHAP1(INODP)
      RHST(INODP) = RHST(INODP) + SHAPI*HSTAR*TFSTAR*SLETH
      DO 95 JCON1 = 1,4
      JCOLP = (JCON1-1)*7 + 2
      DO 95 JCON2 = 1,2
      JNODP = (JCON1-1)*2 + JCON2
      JCOLU = (JCON1-1)*7 + JCON2*4-3
      JCOLT = (JCON1-1)*7 + JCON2*3 + 1
      SHAPJ = SHAP1(JNODP)
      TERM = HSTAR*SHAPI*SHAPJ*SLETH
      FLUMX(IROWT,JCOLT) = FLUMX(IROWT,JCOLT) +TERM
95    CONTINUE
75    CONTINUE
150   CONTINUE
26    CONTINUE
C-----------------------------------------------------------------
C        INCORPORATING NORMAL VELOCITY BOUNDARY CONDITION
C-----------------------------------------------------------------
      ANGLN(82) = 90.0
      DO 80 II = 1,3
C--------------------------------------------------------------------
C   EVALUATING DIRECTION OF NORMAL AT EACH NODE
C-------------------------------------------------------------
      GOTO(100,110,120,130),KSIDE
100   CONTINUE
      XEQIV = FLOAT(II) - 2.0
      YEQIV = -1.0
      ITEMP = 1
      RTEMP = +1.0
      GOTO 140
110   CONTINUE
      XEQIV = 1.0
      YEQIV = FLOAT(II)-2.0
      ITEMP = 2
      RTEMP = 1.0
      GOTO 140
120   CONTINUE
      XEQIV = -FLOAT(II)+2.0
      YEQIV = 1.0
      ITEMP = 1
      RTEMP = -1.0
      GOTO 140
130   CONTINUE
      XEQIV = -1.0
      YEQIV = -FLOAT(II)+2.0
      ITEMP = 2
      RTEMP = -1.0
```

```
      CALL SHAPE8(DERX,SHAPN,XEQIV,YEQIV)
C-----------------------------------------
C     EVALUATING SLOPE AT EACH NODE
C-----------------------------------------
      INODP = 2*(KSIDE-1) +II
      IROWU = 7*(KSIDE-1) + M(II)
      IF(INODP.GT.8) INODP = INODP - 8
      IF(IROWU.GT.28) IROWU = IROWU - 28
      KPOIN = IABS(LNODS(KBEL,INODP))
      IROWV = IROWU+NODFM(KPOIN) - 2
      IROWT = IROWV +1
      IF(KSURF.EQ.5) ANGLN(82) = 20.0
      ANGLE = ANGLN(KPOIN)
      ANGLE = ANGLE*PI/180.0
      COSLX = RTEMP*COS(ANGLE)
      COSN(II) = COSLX
      IF(ABS(COSN(II)).LT.1.0E-4) COSN(II) = 0.0
      COSLY = -RTEMP*SIN(ANGLE)
      SINN(II) = COSLY
      IF(ABS(SINN(II)).LT.1.0E-4) SINN(II) = 0.0
      DO 90 JCON1 = 1,4
      JCOLP = (JCON1-1)*7 + 2
      DO 90 JCON2 = 1,2
      JNODP = (JCON1-1)*2 + JCON2
      JCOLU = (JCON1-1)*7 + JCON2*4-3
      JCOLV = JCOLU +3 - JCON2
      JCOLT = JCOLU +4 - JCON2
      SHAPPJ = SHAPN(JNODP)
C-----------------------------------------------------------
C   DETERMINING APPROPRIATE MOMENTUM EQN. FOR NORMAL VEL. BC.
C-----------------------------------------------------------
      IF(ABS(COSN(II)).GE.ABS(SINN(II))) GOTO 55
      FLUMX(IROWV,JCOLU) = COSN(II)*SHAPPJ
      FLUMX(IROWV,JCOLV) = SINN(II)*SHAPPJ
      FLUMX(IROWV,JCOLT) = 0.0
      IF(JCON2.EQ.2) GOTO 56
      FLUMX(IROWV,JCOLP)= 0.0
56    RHSV(INODP) = 0.0
      GOTO 90
55    CONTINUE
      FLUMX(IROWU,JCOLU) = COSN(II)*SHAPPJ
      FLUMX(IROWU,JCOLV) = SINN(II)*SHAPPJ
      FLUMX(IROWU,JCOLT) = 0.0
      IF(JCON2.EQ.2) GOTO 67
      FLUMX(IROWU,JCOLP) = 0.0
67    RHSU(INODP) = 0.0
90    CONTINUE
80    CONTINUE
C-------------------------------------------------------------
C     ASSEMBLING GAUSS PT. INTEGRALS IN GLOBAL RHS VECTOR
C-------------------------------------------------------------
43    DO 200 II = 1,3
      NOD = NODE(II)
      KPOIN = IABS(LNODS(KBEL,NOD))
      ITOTU = NADFM(KPOIN)
      ITOTV = ITOTU + NODFM(KPOIN) -2
      ITOTT = ITOTU + NODFM(KPOIN) -1
      EQRHS(ITOTU) = EQRHS(ITOTU)+RHSU(NOD)
      EQRHS(ITOTV) = EQRHS(ITOTV)+RHSV(NOD)
      EQRHS(ITOTT) = EQRHS(ITOTT)+RHST(NOD)
200   CONTINUE
88    RETURN
      END
<cms_26>11
```